Towards Securing Networks of Resource Constrained Devices: A Study of Cryptographic Primitives and Key Distribution Schemes

A Thesis
Presented to
The Academic Faculty

by

Kevin S. Chan

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
December 2008

Towards Securing Networks of Resource Constrained Devices: A Study of Cryptographic Primitives and Key Distribution Schemes

Approved by:

Professor Faramarz Fekri, Chair, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor James McClellan
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Steven McLaughlin
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Yajun Mei
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor John Copeland
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: 19 August 2008

In honor of my parents,

Betty and Tai

ACKNOWLEDGEMENTS

First of all, I thank God for providing me with strength, perseverance and focus, for He has been a constant in my life.

I would like to first thank my advisor Dr. Faramarz Fekri for his advisement and guidance over the years. I hope to make him proud of me being his student in my future academic and professional endeavors. I would like to thank my thesis committee for their time, effort, and suggestions for my research. I also thank all of CSIP for the academic and social environment that it has provided to get work done as well as to not get things done. I am grateful to past and present inhabitants of GCATT 324 and Centergy 5204, my home for the last few years. I would like to thank Dr. Fekri's research group (a.k.a. the wavelet coding and cryptography laboratory (WCCL), the wavelet encoding and encryption theory, and most recently the information processing, communications and security research lab (IPCAS) and everyone's input and support through the years. I thank SPG/ENEWS/TEWD/USNRL and all other acronyms for their support during my internship there. Specifically, I thank David Tremper and Dr. Larry Schuette for their guidance and encouragement. Individuals from Georgia Tech that I would like to thank are Mina Sartipi and Maneli Noorkami for their presence in the beginning, and thanks for not thinking I was right on that problem in coding theory class. Also, I'm thankful for Majid Fozunbal and Raviv Raich, for their tutelage in the beginning and their passionate introduction to LaTeX. I'm also indebted to Matt Lee and Paul Hong for letting me into your lunch rotation, introducing me to 'worlds' that I would never have seen on my own, and I look forward to seeing you guys soon. I was fortunate to have Will Leven show up to have someone to hang out with, and Nicolas Gastaud for a cubicle antics partner-in-crime. Also, I'm also glad to have had Badri Vellambi around for his support and commiseration during the last year or so.

I'm also indescribably thankful for people in Atlanta that i've been able to know, giving me that necessary break from school. I'd like to thank Mike Sun for his friendship and

having a fellow Michigan football fan around. Also, I'd like to thank Jenna Fu for her support and encouragement throughout the past couple years, for which I am eternally grateful. The time in Atlanta that I've been able to spend with my cousins Barry and Rosa has been priceless. I'm fortunate that the were in Atlanta to help me adjust to life in the south. To see Megan and Daniel enter their lives, I am overjoyed to see the family grow and to be the kid's human playground. Perhaps the joy got a little more crazy once the other relatives arrived.

I'm also grateful for my friends and life-long relationships that I've found in the other places I've lived. I hope I can maintain contact with my friends at home in Troy, Michigan. I also hope I can do the same with my friends I met at CMU. Everyone has truly been a blessing to my life, and I hope I've done some part to return the favor.

My family has been a solid force that I cannot thank enough. Jason and Margy, this feels like a toast, but so much drier. Finally, I'd like to thank my parents for their unending support encouragement, and prayers to that end. I hope that I have made them proud. I consider them to be my ultimate mentors. I will never know a thing about cascade impactors, but perhaps key predistribution meets the same fate.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis studies security issues pertaining to the rapidly growing area of wireless networks. These networks afford many benefits compared to their wired counterparts in terms of their usability in dynamic situations, mobility of networked devices, and accessibility to hostile or hazardous environments. However, these networks create unique challenges that must be addressed in order for them to be effective. The devices used in these networks are generally assumed to be limited in resources such as energy, memory, communications range, and computational ability. Additionally, these networks can operate in remote or hostile environments, placing them in danger of being damaged upon deployment or compromised by some malicious entity. This thesis addresses some of these issues in an attempt to increase the security of these networks while still maintaining acceptable levels of networking performance and resource usage.

We investigate new methods for data encryption on personal wireless hand-held devices. An important consideration for resource-constrained devices is the processing required to encrypt data for transmission or for secure storage. Significant latency from data encryption diminishes the viability of these security services for hand-held devices. Also, increased processing demands require additional energy for each device, where both energy and processing capability are limited. Therefore, one area of interest for hand-held wireless devices is the ability to provide data encryption while minimizing the processing and energy overhead as a cost to provide such a security service. We study the security of a wavelet-based cryptosystem and consider its viability for use in hand-held devices.

This thesis also considers the performance of wireless sensor networks in the presence of an adversary. The sensor nodes used in these networks are limited in available energy, processing capability and transmission range. Despite these resource constraints and expected malicious attacks on the nodes in the network, these networks require widespread,

highly-reliable communications. Maintaining satisfactory levels of network performance and security between entities is an important goal toward ensuring the successful and accurate completion of desired sensing tasks. However, the resource-constrained nature of the sensor nodes used in these applications provides challenges in meeting these networking and security requirements. We consider link-compromise attacks and node-spoofing attacks on wireless sensor networks, and we consider the performance of various key predistribution schemes applied to these networks. We investigate the resilience of networks with regard to different adversarial attacks. Furthermore, we propose new key predistribution techniques to improve the security of wireless sensor networks.

CHAPTER I

INTRODUCTION

Wireless networks have become vital components in the telecommunications industry both in current applications as well as in research and development efforts. They have been found to be suitable for use in a multitude of applications involving communications and the networking of information between various entities. The wireless aspect of these networks affords many benefits compared to their wired counterparts in terms of their usability in dynamic situations, mobility of networked devices, and access to hostile or hazardous environments. However, the nature of these devices also creates unique challenges that must be dealt with in order to achieve adequate performance for their intended purposes. When considering wireless devices, it is generally assumed that they are limited in resources such as energy, memory, communications range, and computational ability. Additionally, these networks will operate in remote areas in certain scenarios, putting them at risk of being compromised or destroyed by an adversarial entity. Depending on the type of network and deployment environment, the priority of these limitations varies. We are interested in networking situations that pertain to commercial and military applications for wireless networks. We have considered various networking scenarios and addressed the limitations of these environments to maximize the performance of a network employing specific security services.

In terms of commercial applications for wireless devices, the primary challenges stem from the limitations in computational performance and available energy. Personal data assistants (PDAs) and cellular phones are two widely-used examples of resource-constrained devices used in wireless networks. The deployment environment presents a scenario where conserving energy for the processing and transmission of data is of paramount importance. Processors in these devices are significantly limited compared to those found on wired networks. The use of these devices is likely to be in a mobile situation, presenting a situation

where having boundless energy is not feasible. An ever-present challenge in wireless devices for consumer applications is to optimize computational performance and other energy intensive functions to maximize battery lifetimes.

In addition to limitations of the available processing and energy, another factor that places a burden on these wireless devices is the security of information. The environment in which these devices operate is an open environment, where any data being transmitted is vulnerable to attacks from malicious entities. Any data sent on wireless channels is subject to potential eavesdropping, which eliminates the privacy of information being stored or transmitted. As in any security application, securing systems or networks is best carried out in a multi-level fashion [94]. Providing a single level of fortification against adversarial attacks is not desirable nor is it possible to defend against all varieties of adversaries. The necessity of employing any security service on these devices increases the strain on the limited processing and available stored energy. For instance, to guarantee the privacy of information, data encryption is one solution that is available. Having this service improves the overall security of the network, but the network incurs transmission and processing overhead. Given the existing limitations with the processors and stored energy, the increase in bit-rates and processing from the added security services further stresses the devices of the network. The challenge that we consider is minimizing these incurred overheads while still providing certain levels of security in the network.

There are a great number of proposed military applications for wireless networks. Some applications are made of large-scale networks of networks, where devices with unbounded resources communicate over long distances using high bandwidths. With a large amount of information being transferred, high accuracy and low transmission delay are of utmost importance. Applications of wireless networks for tactical purposes can also be found on the very small-scale spectrum of devices, where networks of small radio-frequency (RF) devices are deployed to perform a specific mission. Collaboration of a large number of these devices attains similar capabilities as smaller networks of more capable devices. The attractive features of the small devices are their flexibility in their deployment configuration and the wide variety of potential applications. Since it is likely that these tactical wireless networks

will operate in adverse environments where it is infeasible for other types of networks, harsher networking restrictions and requirements are necessary.

The wireless sensor network is one variety of a tactical wireless network, which is a network of small radio-frequency (RF) sensor nodes that is deployed into a remote environment. These networks are deployed to gather data and relay the information back to a central base station for further processing and decision-making. Each constituent node is limited in transmission range and available energy, and it does not necessarily have direct contact with the base station. Thus, the nodes have to communicate with the base station in a multi-hop fashion through other sensor nodes. Their remote deployment furthers the restriction of the available energy as replacing or recharging the stored energy in nodes is not viable. In addition to the transmission range and energy limitations, remote deployment results in a physical vulnerability of the nodes in these networks. With the flexibility of deployment and potentially boundless capabilities of such networks, there have been many proposed applications for wireless sensor networks [3, 36]. However, the networking performance of wireless sensor networks in response to their physical vulnerability must be addressed in order to successfully deploy these networks for any application.

Many proposed applications for wireless sensor networks call for the nodes to be randomly deployed or dispersed into a field. Upon deployment, some nodes may be damaged or functional but not able to communicate with any other node because of its limited communication range. The physical vulnerability of the nodes presents a challenge that the network must still function properly even after some of the nodes are captured by an adversary. Additionally, the adversary may attempt to degrade the performance of the network or even try to use the captured nodes for its own malicious actions. The conflicting requirements of acceptable performance and security are necessary properties of these networks; however, optimizing these properties is the challenge.

This work considers the performance of resource-constrained devices by investigating two separate problems within the area of security for wireless networks. Figure 1.1 is an illustration of a wireless network, and it highlights the two main areas of wireless network research that we have investigated: link security and network security. These two security-related

issues have been considered separately in the context of resource-constrained devices. First, we examine the performance of a new cryptosystem for use in commercial applications to provide data encryption. This work investigates link security for wireless communications between hand-held devices. Second, we investigate key management schemes for wireless sensor networks and examine the performance of the network in the presence of an adversary. This work evaluates the viability of security services of the network on a global scale.

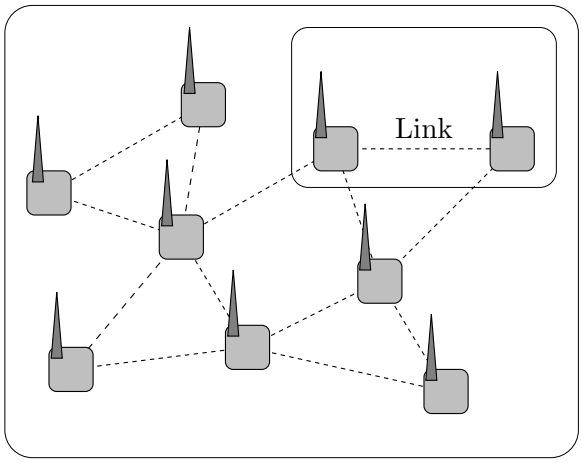


Figure 1.1: Illustration of a network of wireless devices.

1.1 Link security

In Part 1 of this thesis, we investigate new methods for data encryption on personal wireless hand-held devices. An important consideration for resource-constrained devices is the processing required to encrypt data for transmission or for secure storage. Significant latency from data encryption diminishes the viability of these security services for hand-held devices. Also, increased processing demands require additional energy for each device, where both energy and processing capability are limited. Therefore, one area of interest for hand-held wireless devices is being able to provide data encryption while minimizing the processing and energy overhead as a cost to provide such a security service. The research will focus on wavelet-based schemes to address the constraints we have mentioned.

Chapter 2 presents background information for classical cryptosystems, traditional cryptanalytic techniques, and concepts fundamental to wavelet-based encryption. Chapter 3 presents an analysis of the Wavelet Block Cipher (WBC), a private key cryptosystem based on the finite-field wavelet. The encryption and decryption are performed by the synthesis and analysis banks of the nonlinear finite-field wavelet transform, whose filter coefficients are the secret keys. The wavelets operate over $\mathbb{GF}(256)$ and the structure of the cryptosystem also includes a nonlinear device that performs a mapping of the field elements to their inverse in the field. In this thesis, the security and computational complexity of WBC are studied and compared to both the Data Encryption Standard (DES) [70] and Advanced Encryption Standard (AES) [22]. First, we assess WBC from a cryptanalytic standpoint. The security is tied to the wavelet basis function and to the nonlinear function within the WBC encryption round. Additionally, we study the security of the block-cipher wavelet cryptosystem in response to classical attacks and attacks specific to the wavelet structure. In particular, the cryptanalytic methods chosen to be studied with regard to WBC are the divide-and-conquer attack, the interpolation attack, and attacks using properties of the discrete Fourier transform. We show that chosen ciphertext attacks (CCA) on WBC reduce to the problem of solving sets of nonlinear equations over finite-fields. While considering classical and algorithm-specific attacks, the computational complexity to perform these attacks is compared to the exhaustive key search. If a particular attack has a complexity greater than the exhaustive key search, then the cryptosystem is considered to be resilient to that particular attack. The contribution of this thesis to WBC encryption for resource-constrained hand-held devices focused on security analysis.

1.2 Network security

Part 2 of this thesis considers the performance of wireless sensor networks in response to the addition of security services and the presence of an adversary. These networks require widespread, highly-reliable communications even after malicious attacks. Maintaining satisfactory levels of network performance and security between entities is an important goal toward ensuring the successful and accurate completion of desired sensing tasks. However,

the resource-constrained nature of the sensor nodes used in these applications provides challenges in meeting these networking and security requirements. The primary limitation of these devices is the available energy for each of the nodes. This constraint limits the amount of processing and communications a node can complete before running out of energy. Because of the remote nature of these deployments, it is not viable for the energy supplies to be recharged or replaced. Additionally, the communications range is limited, forcing each node to communicate through other nodes to transmit information to a particular destination.

Every node cannot directly communicate with a trusted third party (TTP), so implementing specific security services becomes a challenge. For instance, public key infrastructures (PKI) [85, 97] provide a solution to distribute session keys for data encryption, but without a TTP in direct communication with each node, this is not possible. Communicating with the TTP over multiple-hops renders the network susceptible to the man-in-the-middle attack, an attack that compromises all perceived security gained from the PKI. A recently proposed solution to the key distribution problem for wireless sensor networks is key predistribution. In this situation, each node is given key information prior to deployment. After deployment, nodes establish secure communications links with neighbor nodes based on shared key information. As stated previously, the physical vulnerability of the nodes allows an adversary to gain access to key information in the network. A goal of the key predistribution scheme is to minimize the capability of the adversary while still providing an acceptable networking performance.

The capability of the adversary depends on the ability of the adversary in addition to the amount of resources compromised. In this thesis, we consider adversaries of various strengths and determine to what extent a malicious attack has harmed the performance of the network. For each of these attacks, we measure the increased effectiveness of the adversary as a function of the fraction of the network that the adversary has captured. Based on the key predistribution scheme used in the wireless sensor network, the resources compromised vary with the number of nodes compromised. With regard to network security, we consider two types of attacks: link compromise and node spoof. An adversary wanting to gain access to information being encrypted and transmitted in the network will attempt

to compromise the links in the network. Given a compromised link, the adversary is able to obtain any information transmitted through the once-secure link. This attack is realized by obtaining the keys used to create the targeted links, by capturing other nodes in the network that have the desired keys. We also consider an adversary that is attempting to insert a spoofed node into the network and not be detected as an adversarial node. If this is possible, the adversary can use the spoofed node to perform malicious tasks within the unsuspecting network.

Within the scenario of adversarial attacks on wireless sensor networks, we analyze several aspects of the performance of the network. Chapter 2 presents background material relating to wireless sensor networks and proposed security services for these networks. The key predistribution schemes that we primarily consider in this work are the base scheme or the random key predistribution scheme (QCOMP, BASE) and the multivariate key predistribution scheme (MKPS). We also consider variants of these schemes.

In Chapter 4, we consider the connectivity property of wireless sensor networks and key predistribution schemes. Here, we explore the link-compromise attack and establish a metric measuring the resilience of the connectivity of a network to the link-compromise attack. The resource cost of implementing key predistribution schemes on wireless sensor networks is also explored. We investigate the effect of adversarial attacks on the resilience of secure links generated by key predistribution schemes for wireless sensor networks. These network properties are examined by determining the communication range required of each node to provide global connectivity to the network. We develop a resiliency-connectivity metric, which is used to compare the resilience of networks against the link-compromise attack in terms of the connectivity property of networks.

Chapter 5 investigates the performance of wireless sensor networks and key predistribution schemes for properties other than connectivity. We add the temporal aspect in our analysis of network resilience of node compromise attacks. In our consideration of network properties involving the temporal aspect in wireless networks we show the following results. We measure the average packet latency and packet reliability in a data-gathering scenario.

Using results of latency measurements for different parameters, the throughput and capacity of these networks is determined. Similar to the results in Chapter 4, the resilience of networks to link-compromise attacks in terms of latency and maximum throughput are examined.

Our contributions in Chapter 6 pertain to the node-spoofing attack. In contrast to the adversarial ability when considering link-compromise attacks in sensor networks, the node-spoofing attack requires an increased amount of resources but provides a more capable adversary. We consider the strength of key predistribution schemes against the node-spoof attack. Additionally, we design key predistribution schemes that possess increased resilience to the node-spoofing attack compared to existing schemes. We also consider the node-spoofing attack with several adversarial models that use attackers of varying capability.

CHAPTER II

BACKGROUND

This chapter provides an overview of the topics pertinent to the work discussed in the remainder of this work. Section 2.1 establishes some commonly used notation and definitions for the two main research areas. In Section 2.2, we discuss the major background and current developments within block cipher cryptosystems and cryptanalytic techniques. We also provide some fundamental principles of filter banks, which is necessary for our work in Chapter 3. Section 2.3 contains an overview of properties of wireless sensor networks, several security services, and common adversarial models.

2.1 Notation and Definitions

This section contains an overview of the notation and definitions used throughout this work. Additional terms may be defined in individual sections as necessary. First, we define some notation regarding wavelet-based cryptography that is presented in Chapter 3. Second, we define some notation and network models used for the study of security and network performance of wireless sensor networks, which is used in Chapters 4, 5, and 6.

2.1.1 Cryptography: terms and usage

We establish common notation for the investigation of cryptosystems for hand-held devices. In this work, we consider cryptographic primitives, specifically block ciphers. The encryption of plaintext, P, with secret key, K, yields ciphertext, $C = E_K(P)$. When considering block ciphers, the input and output are a block of x contiguous bits. Some common terms and notation related to encryption using block ciphers are defined in Table 2.1.

Additionally, we consider a new cryptographic primitive and present the recently proposed wavelet block cipher, which is comprised of elementary encryption and decryption blocks based on the finite-field wavelet. We define these blocks and their usage in Table 2.2

Table 2.1: Common cryptographic terms and usage

$C^{(m)}$	Ciphertext block m.
$P^{(m)}$	Plaintext block m.
$E_K(P)$	Encryption of P with key K.
$D_K(P)$	Decryption of P with key K.
C_{even}	The even indices of C, where $C = C_{even}\|C_{odd}$.
C_{odd}	The odd indices of C, where $C = C_{even}\|C_{odd}$.
$[C^(m)C^{(m+1)}]$	Consecutive blocks of ciphertext.

We consider arithmetic in the Galois Field, $\mathbb{GF}(256)$ a finite-field of order 256 and characteristic 2. For any elements in $\mathbb{GF}(256)$, addition and multiplication are closed operations. Also, for any element α in $\mathbb{GF}(256)$, we define $Inv(\alpha) = \alpha^{-1}$ for any non-zero α, where $Inv(0) = 0$. Within each of the encryption and decryption structures for WBC, there are elementary encryption and decryption blocks, which have matrix representations $T^{(i)}$'s and $F^{(i)}$, respectively. The index (i) denotes unique occurrence of that elementary block in the cryptosystem. For each of the elementary blocks in WBC, the filter coefficients $e_{oo}^{(i)}(n)$ are defined by the secret key. Several terms specific to the wavelet-based encryption are defined in Table 2.2.

Table 2.2: Wavelet-based encryption terms and usage

$T^{(i)}$	The ith Elementary encryption block.
$F^{(i)}$	The ith Elementary decryption block.
$F^{(ab)}$	The cascade of the ath and bth elementary decryption block, $F^{(ab)} = F^{(a)}F^{(b)}$.

Additionally, in the construction of the wavelet-based cryptosystem, one-circulant and two-circulant matrices are employed. These are defined as follows.

- We denote by π the permutation matrix defined by

$$\pi = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \tag{2.1}$$

  It is clear that if π has dimension M, then $\pi^T = \pi^{-1} = \pi^{M-1}$.

- We denote one-circulant matrices by 1-circ(a). A one-circulant matrix is defined by its first row $a = [a_0, a_1, \ldots, a_{M-1}]$, and the ith row is equal to $(\pi^{i-1}a^T)^T$. In other words, the ith row is equal to the left-to-right cyclic shift of the vector a by $(i-1)$.

- We denote two-circulant matrices by 2-circ(a). A two-circulant matrix is defined by its first row $a = [a_0, a_1, \ldots, a_{M-1}]$, and the ith row is equal to $(\pi^{2i-2}a^T)^T$.

2.1.2 Wireless sensor networks: notation and models

With respect to our work regarding to wireless sensor networks, we use the following notation and definitions. The network consists of n nodes randomly distributed into a field of unit area. Each node has communication radius $r(n)$, which in this work we consider a uniform radius. The set of nodes that a node can directly communicate with is its neighborhood. In the data gathering scenario, a base station is located in the center of the unit area at coordinate $(0.5, 0.5)$. Some commonly used terms within these networks are listed in Table 2.3.

With regard to the node-compromise attacks, we consider x compromised nodes or p_c, the fraction of nodes in the network which are compromised. For matters of analysis, p_e is the probability of establishing a link and p_{sf} is the probability of sensor properly functioning. We define the the probability of secure key establishment p_k, and the probability of a link being compromised by the adversary to be p_ℓ. This thesis investigates these network properties further and defines several parameters within these two properties. To facilitate analysis, we consider several random graph models to represent wireless sensor networks. They are listed in Table 2.4. Each of these graphs represent a specific situation of the wireless sensor network problem that we address.

Table 2.3: Wireless sensor network notation and usage

v_i	Node with identity i
$N(v_i)$	Neighbor hood of node i
k_{v_i}	Key ring of node i
$k_{v_i}^j$	The jth key in the key ring of node i
$k_{ij} = k_{ji}$	Shared keys between node i and j
P	Key pool size
P'	Compromised key pool
$P_{N(v_i)}$	Key pool of neighborhood of v_i
λ	Key threshold
$\#(k)$	Number of occurrences of key k in the remaining nodes

Table 2.4: Random graph models for wireless sensor networks

$G(n, p)$	Erdös – Rényi random graph
$G(n, r)$	Geometric random graph
$G(n, r, p_k)$	Secure connectivity random graph
$G(n, r, p_k, p_c)$	Compromised secure connectivity random graph

These random graph models provide representations that are usable approximations for the behavior of wireless sensor networks. The Erdös–Rényi random graph, $G(n,p)$, is a graph of n nodes and there is an edge between any two nodes in the network with probability p. The geometric random graph, $G(n,r)$, is a graph of n nodes and there is an edge between any two nodes that are within distance r. The secure connectivity random graph, $G(n,r,p_k)$, is similar to the geometric graph, except that there is an edge between any two nodes that are within distance r with probability p_k. Also, the compromised secure connectivity random graph, $G(n,r,p_k,p_c)$, represents a secure connectivity random graph where a fraction p_c of the nodes are removed from the network. In Chapters 4 and 5, we see that the parameter p_c has an effect on the rate of link compromise rate, p_ℓ.

2.2 Cryptographic primitives: block ciphers

In 1949, Shannon began the discussion of information secrecy [91]. In the classical description of the security we are attempting to achieve, we imagine that two entities, Alice and Bob, wish to exchange messages through an insecure channel in such a way that an adversary Oscar cannot understand the communication. The security service described here is one of data secrecy, which is carried out through use of cryptographic techniques. Two categories that cryptographic schemes can be broken down into are public key cryptosystems and private key cryptosystems.

In general, the public key systems are used to private key systems generally use to establish secret keys and then use private key systems for the encryption of data to be transmitted. The difference in the computational complexity between public and private key systems is significant enough to warrant such use. Assuming session keys are already established between two devices, there still is an accrued overhead when encrypting with private key cryptosystems. Compared to sending data in the clear, these cryptosystem inherently require an overhead of extra computations and energy usage. Minimizing the overhead of the encryption schemes is desirable in the case of hand-held devices. Two common private key cryptosystems that provide acceptable security and minimal computational complexity are the Advanced Encryption Standard [22] and the Data Encryption Standard

DES [70], referred to as AES and DES, respectively.

The fundamental building block for security is the cryptographic primitive. Other cryptographic functions are built from this building block, such as block ciphers, stream ciphers, hash functions, and digital signatures. These primitives are perceived to be nearly one-way functions without knowledge of the secret key. With the secret key, they are easily invertible. Here, we focus on the block cipher, a private key cryptosystem that uses a secret key K to operate on fixed blocks of input/plaintext P and produce an output/ciphertext C of the same length ($E_K(P) = C$). These systems are also considered to be symmetric, in that the decryption of the ciphertext takes the same key and uses the same encryption functions to recover the plaintext.

Data Encryption Standard (DES): The Data Encryption Standard, commonly referred to as DES, is a cryptosystem that was introduced for use in the 1970s. Its basis is the Feistel cipher, on which many encryption algorithms are based [73, 89, 68, 10, 1, 86]. The Feistel cipher consists of a simple non-linear function that is repeated many times to achieve its security. The security of DES is based on a nonlinear operation called the s-box.

Advanced Encryption Standard (AES): A more recent development in the cryptographic community is the development of the Advanced Encryption Standard, referred to as AES. After an open call and competition throughout the cryptographic community, a new encryption standard called Rjindael was selected as the new standard, AES. The encryption and decryption are done by repeatedly executing several simple functions over a specified number of rounds on 128-bit blocks. The key is also 128 bits although Rjindael is capable of various other encryption block sizes. The `State` is the working ciphertext and `RoundKey` is the key used in each round that follows a publicly known key schedule such that each round key is derived from the original key K. One round of AES consists of the following operations on `State`: `ByteSub(State)`, `ShiftRow(State)`, `MixColumn(State)`, and `AddRoundKey(State, RoundKey)`. The transformations in AES can be represented as matrix operations. The arithmetic is done in $\mathbb{GF}(2^8)$.

2.2.1 Classical cryptanalytic techniques

According to *Kerckhoffs' principle*, the security of these cryptosystems is based on the adversary only not knowing the secret key [54]. It is assumed that the structure of the algorithm is publicly known, so the adversary's goal is simply to determine the secret key. Of paramount importance to any cryptosystem is the analysis of the security of these algorithms. The necessity for novel and more complex cryptographic algorithms is rooted in the cryptanalysis of these algorithms. As new cryptographic primitives are introduced, so too are new cryptanalytic techniques. We outline several desirable properties of cryptographic algorithms, those that have exhibited strength against known attacks. Also, we include a description of some of the known attacks on cryptosystems.

There are several attack types for the cryptanalysis of these algorithms. There are chosen plaintext attacks (CPA), where the adversary is able to select plaintext and observe the output ciphertext for a particular instance of a cryptosystem. Conversely, there are also chosen ciphertext attacks (CCA). Less powerful or able attacks are the known plaintext and ciphertext attacks (KPA, KCA), where pairs of plaintext and ciphertext are available. We note that for block ciphers, the variants of ciphertext and plaintext attacks are similar because of the structure of the encryption.

There are several desirable properties of cryptographic algorithms that can be used to prove resilience against some known methods of attack. Shannon introduces confusion and diffusion properties [91]. In these systems he talks of information secrecy being attained with a complex mixing of the key and the plaintext and a dependence of every bit of the ciphertext on every bit of the plaintext. For instance, AES claims full diffusion in two rounds of encryption. Webster furthered the idea by introducing the Strict Avalanche Criterion [98], which states that flipping one bit in the plaintext results in each bit in the ciphertext to change with $p = 0.5$. Also, it is desirable for cryptosystems not to have weak keys. In these situations, encryption with weak keys results in a system with some undesirable properties. For instance in DES, the keys $\{[0_{64}], [1_{64}], [1_{32}|0_{32}], [0_{32}|1_{32}]\}$ result in the property where encrypting twice with the same key recovers the plaintext, $E_K(E_K(P)) = P$. Weak keys can additionally cause the cryptosystem to exhibit poor security.

We now introduce the major cryptanalytic techniques that gave rise from a security analysis of DES. Attributed to Biham and Shamir, differential cryptanalysis [6] was found to provide a vulnerability in DES. Matsui is credited for developing Linear cryptanalysis [67]. Linear and differential attacks [6, 67] are two classical cryptanalytic attacks that have exposed security vulnerabilities in many cryptographic algorithms. The viability of these attacks is based on the presence of strong characteristics. Occurring with a certain probability, a characteristic is a relationship of plaintext, ciphertext, and key bits that spans one or more rounds of the elementary round function. Attacks are formed by concatenating these characteristics to establish a relationship among bits of the plaintext, ciphertext, and key bits of the full cryptographic algorithm.

The differential characteristic considers the distribution of the output differential (here the XOR operation is considered) of two decrypted plaintexts corresponding to two chosen ciphertexts with a fixed differential XOR. The non-linear functions with the highest degree of non-uniformity in the distribution among the ciphertext XOR pairs are vulnerable to a differential attack.

To define the differential characteristic, let there be two plaintexts P, P^* with a bitwise XOR of P' ($P = P \oplus P^*$) and two ciphertexts C and C^* with a bitwise XOR of C' ($C = C \oplus C^*$). A strong characteristic is defined to be the pair of P' and C' that occurs with maximum probability, p, for any K. With P', there are 2^N pairs of P, P^*, where N is the block length, that have a bitwise XOR of P'. Encrypting each P, P^* with K, results in $E_K(P)$ and $E_K(P^*)$. We then determine $C \oplus C^* = E_K(P) \oplus E_K(P)$ and find the C' that occurs most often for any choice of K. The strong differential characteristics occur with maximal probability, p.

Definition 1. *Differential Characteristic: The choice of* P' *and* $C' \in \{\mathbb{Z}_2\}^N$ *that maximizes* $p = \#\{\ (P \oplus P^*)\ \forall\ C \oplus C^* = C'\}/2^N\ \forall\ K \epsilon \{\mathbb{Z}_2\}^N$.

For linear cryptanalysis, the linear characteristic attempts to take advantage of potential linear structures in the non-linear function. There may exist some correlation between some ciphertext, key, and plaintext bits or symbols. The characteristic attempts to find such a

correlation that may be satisfied with probability, p, with strong bias, either close to zero probability or almost certainty, $|p - \frac{1}{2}| > 0$. For the definition of a linear characteristic, let P_i, C_i, and K_i be the i^{th} bit in a block of the plaintext, ciphertext and key, respectively. The characteristic is a combination of an arbitrary subset of bits of P, C, and K, where $P = D_K(C)$. Let these subsets of P, C, and K be indexed by $\{i_1, i_2 \dots i_m\}$, $\{j_1, j_2 \dots j_n\}$, and $\{r_1, r_2 \dots r_\ell\}$, respectively. The cardinality of the subsets ℓ, m, and n, is arbitrary. The characteristic is the relationship for which the XOR sum over all these bits is maximized. p is the probability that this relationship holds for the choice of C, K, and associated $P = D_K(C)$.

Definition 2. *Linear Characteristic: The fixed subset of indices of P, K, C so that $[P_{i_1} \oplus \cdots \oplus P_{i_m}] \oplus [K_{r_1} \oplus \cdots \oplus K_{r_n}] = [C_{j_1} \oplus \cdots \oplus C_{j_\ell}]$ is satisfied with probability p such that $|p - 1/2|$ is maximized $\forall$ C, K.*

Additionally, with the growing trend of finite-field arithmetic in cryptographic algorithms, attacks considering the mathematical structure of these systems were explored. Such algebraic attacks such as interpolation attacks were found to be viable on some cryptosystems [20, 49]. These attacks are based on the idea that there is a mathematical structure in the encryption, where breaking the cryptosystem reduces to solving a system of equations.

2.2.2 The wavelet transform and multi-rate filter banks

Wavelets and multi-rate filter banks have received a great amount of development within the signal processing community [77]. As shown in Figure 2.1, these filter banks offer perfect reconstruction and low computational complexity. Furthermore, Fekri extends the theory of wavelets and filter banks to provide a theory of wavelet decomposition of sequences defined over finite-fields [37, 39, 38, 40, 41].

In finite-field wavelets and filter banks, all the coefficients in the filters and all of the sample values are taken from a finite field and the arithmetic is carried out in that field. If the field is $GF(p)$, p a prime, then addition and multiplication are defined modulo-p. In fields of the form $GF(p^r)$, where $r > 1$, any number a can be represented by a polynomial

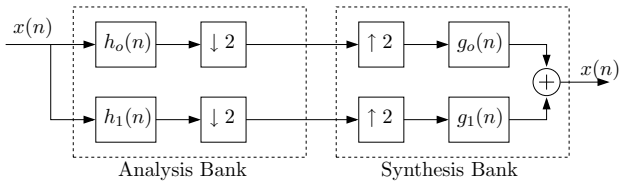


Figure 2.1: Two-channel analysis/synthesis filter bank.

of degree $r-1$ with coefficients from $GF(p)$. Then, addition is defined as addition of polynomials in $GF(p)$, and multiplication is defined to be polynomial multiplication modulo a fixed polynomial $q(y)$. The polynomial $q(y)$ must be an irreducible polynomial of degree r over $GF(p)$ [58].

We describe Figure 2.1 and the perfectly reconstructing property for the wavelet transform along with its representation with matrix transforms. Consider a two-channel maximally decimated filter bank with two analysis filters with impulse responses $h_0(n) = \{h_0(0), h_0(1), \cdots, h_0(N-1)\}$ and $h_1(n) = \{h_1(0), h_1(1), \cdots, h_1(N-1)\}$. In the analysis bank of Fig. 2.1, the operation of filtering periodic signals followed by decimation by a factor of two can be described using two-circulant matrices as:

$$\begin{aligned} y_0(n) &= \sum_{i=0}^{N-1} x(i) h_0((2n-i))_N = (H_0 x)(n) \\ y_1(n) &= \sum_{i=0}^{N-1} x(i) h_1((2n-i))_N = (H_1 x)(n), \end{aligned} \tag{2.2}$$

in which $H_0 = 2\text{-circ}(h_0^R)$ and $H_1 = 2\text{-circ}(h_1^R)$ are $M \times N$ 2-circulant matrices defined by the analysis filters $h_0(n)$ and $h_1(n)$. The operation $(())_N$ denotes a circular shift of length N. In the synthesis bank, the filters have impulse responses $g_0(n) = \{g_0(0), g_0(1), \cdots, g_0(N-1)\}$ and $g_1(n) = \{g_1(0), g_1(1), \cdots, g_1(N-1)\}$. The relationship between the analysis filters and the synthesis filters in

$$h_j((n))_N = g_j((-n))_N \quad j = 0, 1 \quad n = 0, \ldots, N-1. \tag{2.3}$$

The upsampling of periodic signals by a factor of two followed by the filtering operation can be described by (column-wise) 2-circulant matrices $G_0 = [2\text{-circ}(g_0)]^T$ and $G_1 =$

$[2\text{-circ}(g_1)]^T$:

$$\begin{aligned} x(n) &= \sum_{i=0}^{M-1} y_0(i) g_0((n-2i))_N + \sum_{i=0}^{M-1} y_1(i) g_1((n-2i))_N \\ &= (G_0 y_0)(n) + (G_1 y_1)(n). \end{aligned} \tag{2.4}$$

The above formulation holds for the general class of two-channel cyclic filter banks. By using (2.3), these transforms can be described in the following

$$H_j = \begin{bmatrix} g_j(0) & g_j(1) & \cdots & g_j(N-1) \\ g_j(N-2) & g_j(N-1) & \cdots & g_j(N-3) \\ \vdots & \vdots & \vdots & \vdots \\ g_j(2) & g_j(3) & \cdots & g_j(1) \end{bmatrix} \text{ for } \; j = 0, 1. \tag{2.5}$$

Also, because of relation (2.3), the synthesis matrices are the transposes of the analysis matrices:

$$G_j = H_j^T \qquad j = 0, 1. \tag{2.6}$$

From the perfect reconstruction constraint $x = G_0 H_0 x + G_1 H_1 x$, it is possible to determine that

$$[H_0^T \; H_1^T] \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} = I_{N \times N}. \tag{2.7}$$

Finite-field wavelets have been proposed to serve as an elementary building block for the construction of cryptographic primitives. These transforms possess a rich history in signal processing, and these frameworks can be adopted into algorithms in the cryptographic community.

2.3 Security of wireless sensor networks

Security in many applications for wireless sensor networks is paramount. Sensor nodes are deployed into some hostile environments, where there are adversaries present who are attempting to degrade or destroy the performance of its intended use. Security in wireless sensor networks distinguishes itself from other varieties of networks in that there are the resource-constrained nodes and the lack of physical security. It is assumed that nodes are not safe, as an adversary may have direct access to any number of nodes. The security schemes

developed for IP-based networks cannot be used. Universal addressing is not viable because of the scale and limitations of memory and computation on each of the nodes. Generally, nodes are only aware of their neighborhood and perhaps the base station.

We consider the key management problem for wireless sensor networks. First we describe public key cryptosystems and provide justification for why these schemes are infeasible for this variety of networks. We present key predistribution schemes proposed in response to this limitation. We then present an overview of malicious attacks on wireless sensor networks. lastly, we briefly introduce several networking properties and components necessary for analysis of these networks.

2.3.1 Key distribution in wireless networks

In order to provide data privacy to communications throughout the network, it is necessary to securely distribute keys to the nodes throughout the network. We describe several public key cryptosystems and key predistribution schemes that are used by networks to establish session keys.

2.3.1.1 Public key cryptography

Public key systems are used by two parties to establish session keys in a network over an insecure communication channel. Subsequent transmissions can be encrypted using this session key. Each party must consult a trusted third party (TTP) for some public information about the other party in order to execute such algorithms. Two commonly-used algorithms are the Diffie-Hellman key exchange [97] and the RSA algorithm [85]. We briefly describe the construction of these two cryptosystems.

The **Diffie-Hellman key exchange** operates on the multiplicative group of integers modulo p. With p being a large prime and g being a primitive root of p, these two parameters are publicly known. Given a private key for each user in the network, the algorithm is to establish the key $g^{ba} \mod p$ between Alice and Bob is shown in Table 2.5. The security of this key exchange algorithm is in solving for x given y, g, p, where $y = g^x \mod p$. This is the equivalent to solving the discrete-logarithm problem.

RSA is another public key cryptosystem which operates by dealing with large prime

Table 2.5: Diffie-Hellman key exchange between Alice and Bob

Alice		**Bob**
p, g	(public)	p, g
a	(private)	b
$g^a \mod p$	$\rightarrow$	
	$\leftarrow$	$g^b \mod p$
$(g^b \mod p)^a \mod p$		$(g^b \mod p)^a \mod p$
$g^{ba} \mod p$	$=$	$g^{ab} \mod p$

numbers. Again, these public key systems are generally used to generate secret keys for use with faster, more efficient private key cryptosystems. We briefly describe the fundamental RSA algorithm.

Each party in the network creates a private and public key by selecting two large prime numbers p, q. This establishes the modulus for the keys, $n = pq$. The totient function $\varphi(n)$ is then computed by $\varphi(n) = (p-1)(q-1)$. The party then chooses e, where $1 < e < \varphi(n)$. Additionally, d is chosen such that $de \equiv 1 \mod \varphi(n)$. The user then has (d, n) as its private key and (e, n) as its public keys. Alice is able to transmit securely to Bob using his public key (e_b, n). This is described in Table 2.6 as Alice is able to send the message m to Bob. In the key exchange scenario, this message is the session key, $m = k_{ab}$.

Table 2.6: RSA secure communications between Alice and Bob

Alice		**Bob**
e_a, n	(public)	e_b, n
d_a, n	(private)	d_b, n
$c = m^{e_b} \mod n$	$\rightarrow$	
		$c^{d_b} \mod n$
		$m^{e_b d_b} \mod n$
		$m \mod n$

The security of the RSA scheme is found in the difficulty in factoring the product large prime numbers. If an adversary is able to factor p, q from n, then he is able to determine d

for any user given e.

Given the limitations of resource-constrained devices in wireless sensor networks, there are two primary reasons why public key cryptosystems are infeasible. First, it is apparent from the two public key systems introduced that there is a significant computational complexity to establish a session key. For both of these schemes exponentiation is required to encrypt messages. Although there exist methods to reduce the complexity of these operations, these schemes will consume vast amounts of processing and energy. Second, the communication range limitation for the deployed nodes causes some of the nodes not to be in direct contact with the base station. Assuming that the base station acts as the trusted third party for the public key cryptosystems, all nodes would not have direct access to the public information required for these schemes. Attempting to acquire this information through intermediate sensor nodes makes any key established in this way to be susceptible to the man-in-the-middle attack. The adversary launching this attack would be able to decrypt all information transmitted between the attacked nodes. The description of this attack is detailed in Table 2.7, where the adversary Oscar, unsuspecting to Alice and Bob, establishes keys with himself and the two instead of between Alice and Bob using the man-in-the-middle attack.

Table 2.7: Diffie-Hellman key exchange and the man-in-the-middle attack

Alice		**Oscar**		**Bob**
p, g	(public)	p, g	(public)	p, g
a	(private)	o	(private)	b
$g^a \mod p$				
		$g^o \mod p$	$\rightarrow$	
			$\leftarrow$	$g^b mod p$
	$\leftarrow$	$g^o \mod p$	$\leftarrow$	$g^b \mod p$
$(g^o \mod p)^a \mod p$				$(g^o mod p)^b \mod p$
		$(g^o \mod p)^a \mod p$		
		$(g^o \mod p)^b \mod p$		

2.3.1.2 Key predistribution in wireless sensor networks

As public key systems are not considered to be a viable solution for the key management in wireless sensor networks, attention is given to the notion of key predistribution. In this approach, nodes are loaded with key information before deployment. This frees the requirement that the nodes must be in direct contact with the base station. After deployment, each node communicates with its neighbor nodes and attempts to establish session keys. We now describe several approaches to key predistribution for wireless sensor networks.

First, we present two naive approaches to key predistribution: using a global key and using pairwise keys. For networks using the global key approach, every node is preloaded with k_{global}, the same key for each node. After deployment, each node can securely communicate with its neighbors using k_{global}. However, with a single node compromise, all of the secure communication links are compromised. Also, the adversary is able to successfully spoof any identity anywhere in the network. Conversely, the pairwise keys approach presents a different design flaw. At first glance, this approach seems desirable in that each key carries a key for each node (*i.e.* For v_i it has $k_{ij}\ \forall\ v_j \in V$, where V is the set of all possible nodes). A node is able to establish a secure link with every node, but the memory burden is costly for these resource-constrained nodes since they must carry $n-1$ keys.

We consider several other key predistribution schemes that rely on probabilistic techniques to establish keys. Several schemes with different approaches exist [13, 26, 29, 32, 33, 35, 60, 61]. After deployment of these networks, each node is able to establish a pairwise key and communicate securely only with nodes in its communication range and with whom they share adequate key information as specified by the key predistribution scheme. This probability that the two nodes are able to establish a secure link, is defined as p_k. We briefly describe two key predistribution methods, the first proposed scheme and one based on multivariate symmetric polynomials.

Eschenauer and Gligor [35] provide one of the original works on key predistribution schemes for sensor networks. These schemes assume a limited storage ability on each deployed sensor in the network; therefore, they randomly assign a set number of keys m in each node from a key pool of P keys. Two nodes are able to establish a secure link if they

are within another node's communication radius and share a common key. Chan et al. [13] follow up on this scheme to propose a q-composite scheme where q keys are required to establish a link. They derive an expression for the probability that two nodes share i common keys. In this expression, two nodes are randomly given m keys from a key pool of P keys and share i keys with probability

$$P(i) = \frac{\binom{P}{i}\binom{P-i}{2(m-i)}\binom{2(m-i)}{m-i}}{\binom{P}{m}^2}. \tag{2.8}$$

In this scheme, two arbitrary nodes are able to establish a secure link with probability p_k, given by

$$p_{k(\text{QCOMP})} = \sum_{i=q}^{m} P(i). \tag{2.9}$$

In this thesis, we identify terms such as $p_{k(\text{QCOMP})}$ by abbreviating this scheme by QCOMP. We define an instance of a q-composite network by QCOMP(n, q, m, P), which is a unique instance. We also define the BASE scheme to be QCOMP where $q = 1$, and it can be uniquely described by BASE(n, m, P).

Peer Intermediaries for Key Establishment (PIKE) is a proposed key predistribution scheme [12] where node identities are organized in a square grid, where the node identities are represented by coordinate grid points. Each node prior to deployment is loaded with a secret pairwise key with each node that is found in the same row or column of the grid. After deployment, neighboring nodes that do not share a pairwise key are able to use intermediate nodes to establish pairwise keys.

Delgosha [26] proposes a key predistribution scheme based on multivariate symmetric polynomials. Here, each node is given a coordinate in an d-dimensional space, $(i_1, \ldots i_d)$. Let $I_{(j)}$ be the set $(i_1, \ldots i_d)$ minus the jth variable. Each $i_1, \cdots i_d \epsilon [1, m]$, where $s = \lceil \sqrt[d]{n} \rceil$. There are sd symmetric polynomials generated, $f^s_{i_d}(x_1, \ldots x_b)$, one for each enumerated combination of d, s. Based on its identity, the nodes use the associated polynomials. Prior to deployment, each node is loaded with d single variable polynomials, each calculated by evaluating the polynomials corresponding to the value in each dimension in all but one variable. For example, for the node $I = (i_1, \ldots i_d)$, it is given $f^1_{i_1}(x_1, i_2, \ldots i_d)$, ..., $f^s_{i_d}(x_d, i_2, \ldots i_{d-1})$. These are considered to be the key shares of node I. Nodes are able

to establish a secure link with one of its neighbors if their node identities are of Hamming distance of 1 (two nodes IDs differ in only one index). The two nodes are able to establish a shared key by evaluating each key share polynomial at the variable where they differ and combining the d evaluated key shares into a final pairwise key. One can see that the probability that two nodes are Hamming distance 1 from each other, thus being able to create a secure link, is

$$p_{k(\text{MKPS})} = \frac{(s-1)d}{s^d}. \tag{2.10}$$

In this work, we identify terms such as $p_{k(\text{MKPS})}$ by abbreviating this scheme with a MKPS tag. In this work, we define an instance of a network employing the multivariate symmetric polynomial key predistribution scheme by MKPS(n, d, t), which is a unique instance.

The probability of link failure is evaluated, which is stated to be the ability of the adversary to compromise enough nodes to obtain enough shares of every polynomial. The number of required shares to recover d polynomial coefficients is $\lambda(d, t) = \binom{t+r}{r}$ in the case of degree t polynomials, where $r = d - 1$. The probability of a link compromise, p_ℓ, is defined to be

$$p_{\ell(\text{MKPS})} = p_{pr}^{d-1} \tag{2.11}$$

where each of the $d - 1$ common shared polynomials is recovered by the adversary. The expression for the probability of a polynomial recovery, p_{pr}, is

$$p_{pr} = \sum_{i=\lambda(r,t)}^{s^r} \binom{s^r}{i} p_c^i (1 - p_c)^{s^r - i}. \tag{2.12}$$

2.3.2 Attacks on wireless sensor networks

In terms of the malicious behavior expected from the adversarial influences on these networks, the nature of the deployment of the networks assumes a scenario that is lacking in physical security. In this way, the adversary has many possibilities to attack the network. Karlof et al. [52] present several attack models on the routing of sensor networks, including two types of attackers the node-class and laptop-class attackers. The difference of these attack models are in the capabilities of the adversary, which are as follows.

1. *Node-class attacker*: The node-level attacker has the same ability as one of the nodes. The attacker has the same limitations of the regularly functioning node. Its transmission range, processing capability and available energy are all limited.

2. *Laptop-class attacker*: The laptop-class attacker has access to a high bandwidth network, a more powerful transmitter/receiver, unlimited battery life, increased computational ability, and perhaps boundless physical searching capability.

In this thesis, we examine attacks on the network in the form of node compromises. With these nodes and the information accumulated from these nodes, the adversary is able to launch a variety of malicious attacks. The capability of the adversary determines the potential strength of the threat posed to the network. This thesis considers *link-compromise* attacks and *node-spoofing* attacks as a result of node-compromise attacks. With compromised nodes and spoofed nodes, the adversary can carry out various malicious activities.

Of these malicious activities, the adversary may be able to carry out relatively simple attacks such as eavesdropping on communications, replaying any transmissions, and perhaps modifying information and retransmitting packets. There are also some rather severe attacks such as the wormhole, sinkhole and Sybil attacks, which require nodes to be spoofed to carry these attacks out. Other works consider specific attacks on networks; Parno considers defenses against node replication attacks [75]. The case where compromised nodes are spoofed into multiple legitimate node identities, known as the Sybil attack, is considered in [30, 71]. Others attempt to prevent sink hole and wormhole attacks using various approaches [78, 83]. Several works attempt to detect misbehaving of failed nodes in the network with different approaches [50, 64, 93]. There have been other proposed platforms to provide general security services to wireless sensor networks [11, 51, 76, 102]

The attack model that is assumed in many cases, primarily in the case of key management schemes, is that the compromise of a node reveals to all the key information within that node. If the adversary has all keys used to established a secure link in the network it is considered to be compromised. The enemy would be able to decrypt any of these

messages being transmitted without any physical access to either of the nodes in communication. While worst-case scenario design in terms of expected adversary may provide the best capable security measures, it is important to identify the relationship between providing security versus other network parameters.

2.3.3 Properties of wireless sensor networks

This section introduces several properties of wireless networks, all of which have foundations in traditional wired or wireless networks. We consider connectivity, latency and capacity for wireless networks along with routing protocols and medium access protocols.

The analysis of properties of wireless networks begins with studies in random graphs [9, 27]. Gupta and Kumar [43, 44, 99] derive capacity and connectivity results on the asymptotic large-scale random graph $G(n, p)$, where n is the size of the network and p is the probability of establishing a link. However, Pishro-nik [79] proposes a random graph model for use with wireless networks in $G(n, r, f, p_e, p_{sf})$, which considers a finite communication radius and unreliable links and sensors. In these random graphs, f is the distribution of the nodes, p_e is the probability of link error and p_{sf} is the probability of sensor failure. An expression for required communication radius for $G(n, p_{sf}, f, p_e)$ is

$$\lim_{n\to\infty} r(n) \geq \sqrt{\frac{\ln(n)}{\pi n p_e(n) f_{min} p_{sf}(n)}}. \tag{2.13}$$

The required communication range is shown to be a sharp threshold for connectivity and k-connectivity for large networks. Given the network parameters, any $r(n)$ below the threshold will be not connected with high probability, and any $r(n)$ above the threshold will produce a network that is connected with high probability.

There have been studies of other properties of wireless sensor networks to facilitate the flow of packets in the network. Routing within sensor networks has been a significant area of interest as the traditional routing protocols assume an internet-protocol (IP) based architecture [2], which is not viable in sensor networks. Given a multi-hop networking environment along with resource-constrained devices, several different approaches emerge. Heinzelman introduced a clustering [45, 46] approach where nodes become clusterhead nodes

throughout the operation of the network. Other proposed similar hierarchical routing protocols [59, 62, 63], This approach distributes the energy usage so that only a small subset of the nodes are required to transmit with higher power to reach the base station. Also introduced were directed diffusion [47, 48] approaches to routing protocols, which has each node find the node to forward its packets through based on maximizing some global network metric. Also, routing in sensor networks can be achieved with geographic metrics [53, 57, 72, 84, 88, 90, 103] using position information. Medium access control (MAC) protocols in wireless sensor networks have had to deal with the limited communication range of the network in addition to the lack of clock synchronization in the network [82, 92, 100]. These network services are used in our analysis of the performance of sensor networks.

PART I

Link Security

CHAPTER III

SECURITY ANALYSIS OF WAVELET-BASED CRYPTOGRAPHY

3.1 Introduction

As stated in the introduction, we are investigating data-encryption techniques for wireless hand-held devices. While these devices, such as mobile phones, will play a pivotal role in electronic-commerce applications by delivering a range of services anywhere and anytime, they are vulnerable to malicious attacks. Any cellular transmission or conversation is vulnerable because of the wireless transmission medium. Attackers are able to eavesdrop on conversations, steal private data or even pretend to be a legitimate user. If there exists a constant paranoia of insecure communications, users will not be willing to use such technologies. In this chapter, we examine the security and performance of a new cryptosystem based on the finite-field wavelet, which is proposed for use with hand-held devices.

Finite-alphabet processing plays a key role in coding for security. For many years, work on Fourier transforms has had impact on cryptography. For example, several cryptographic analyses have used the discrete-Fourier transform and the Walsh transform [42, 65, 66, 87]. Like the Fourier transform, which has found widespread applications in cryptography, Fekri has proposed to employ finite-field wavelets as elementary building blocks for the construction of cryptographic primitives. Work attributed to Fekri has shown that processing based on a newly developed theory of wavelet transforms over finite-fields provides a framework for new approaches to cryptography. This new theory provides a general wavelet decomposition of sequences defined over finite fields [37, 38, 39, 40, 41]. This is an approach that has a rich history in signal processing for the representation of real-valued signals, but not as prominent in the case of finite-fields. One of the interesting properties of the finite-field wavelet is that it transforms the input message into a form that is very difficult to recover without knowledge of the basis functions.

The motivation behind developing a data encryption technique is to provide an algorithm

for wireless hand-held devices with several properties. This scheme needs to have an efficient implementation, particularly in hardware, as hand-held devices cannot afford complicated hardware. If these algorithms can be implemented on a digital signal processor (DSP), this makes it suitable for use in mobile devices such as cellular phones, digital cameras, and digital camcorders. In addition, we require a flexible design that allows the variation of the key size and also for use in different modes of operation without changing the hardware architecture. Having a simple yet flexible design facilitates the analysis and implementation. Additionally, we require a cryptosystem that is secure against cryptanalytic attacks.

We estimate that the wavelet cryptosystem with key size of 128 runs as fast as AES with key size 128 in software. Our cryptanalytic work suggests that the wavelet cryptosystem remains unbroken against all the known cryptanalytic attacks. Additionally, since wavelets are implemented by digital filters, its execution times are optimized when realized in hardware. In this chapter, we first present a background of the wavelet transform and its adaptation to provide elementary encryption and decryption blocks. From these elementary blocks comes the wavelet block cipher (WBC). These developments with regard to the theory of the finite-field wavelet and their application to cryptography are due to Fekri. This thesis introduces the structure and theory of the wavelet cryptosystem; however, its main contribution is the cryptanalysis of WBC. We analyze the security of the wavelet block cipher by considering cryptanalytic attacks on WBC. Since WBC possesses a strong mathematical structure, we consider algebraic attacks on this cryptosystem. Last, we compare its computational complexity against other widely-used cryptosystems.

3.2 Wavelet structure for encryption and decryption

This section describes the wavelet-based cryptosystem previously proposed by Fekri. An interesting property of the finite-field wavelet is that it transforms the input data to a sequence similar to white noise. More precisely, finite-field wavelets have a decorrelating property. Consider Figure 3.1 where we process an image which has amplitudes ranging from 0 to 255. We can treat this image as a two-dimensional signal in $\mathbb{GF}(256)$. We designed a separable two-dimensional filter bank by cascading two one-dimensional orthogonal filter

banks of filter length eight over the field $\mathbb{GF}(256)$. In a separable two-dimensional wavelet analysis of a two-dimensional signal, the rows and columns of the two-dimensional signal are analyzed separately. In other words, two sets of one-dimensional basis functions are used to represent rows and columns of the two-dimensional signal independently. By this method, the two-dimensional data (signal) is first encoded row-wise and then the resulting data is encoded again column-wise. As it is illustrated in Figure 3.1, the input data is transformed into four sub-sequences that are similar to white noise. This was experimentally observed on several natural images during the development of the finite-field wavelet theory. It is important to note that the finite-field wavelets are quite different from their real-field counterparts. If we used real-field wavelets on the image, we would have had sub-images that contain lowpass and highpass information of the original image. The original image can be perfectly reconstructed by using the finite-field inverse wavelet counterpart. Additionally, as Figure 3.1 may not imply, it is possible to compress the image with any compression method and then treat the compressed output as the input to the encryption algorithm. Figure 3.1 only illustrates the decorrelating property of the finite-field wavelet transform.

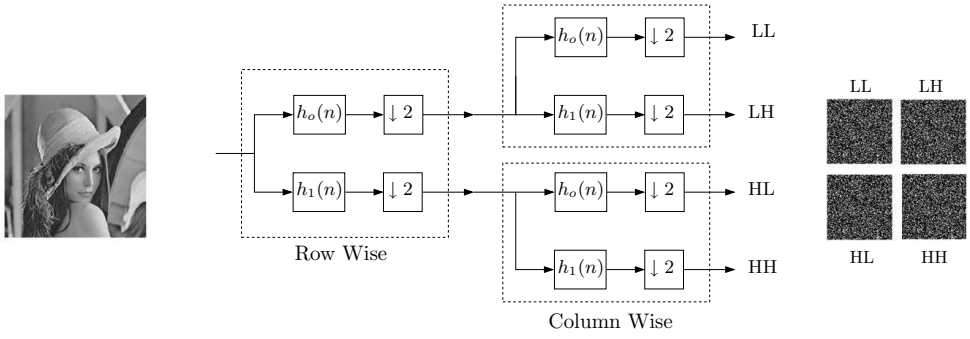


Figure 3.1: Wavelet decomposition of Lenna's image by a two-stage, two-band, orthogonal filter bank over $\mathbb{GF}(256)$.

In addition to this decorrelating property of finite-field wavelets, there are two key properties that are exploited to construct an encryption system where both end users participate in the determination of the key. First, there is a high degree of non-linearity in the wavelet structure by using the lifting scheme. Second, the symmetric property of the polyphase filters are exploited by the transmitter and receiver to construct a shared key. The encryption and decryption are performed by the synthesis and analysis banks of the nonlinear

finite-field wavelet transform, respectively. The wavelet system is determined by the secret and public keys of the users. The security depends on the length of the wavelet basis function and the nonlinearity within the wavelet transform, which operate over $\mathbb{GF}(256)$. The wavelet-based cryptosystem can operate in either stream-cipher [24, 25] or block-cipher [16, 17] modes depending on whether the filter banks perform linear or circular convolution. In this thesis, we study the wavelet-based block cipher. In the following section, we give the details of the wavelet cryptosystem.

3.2.1 Linear blocks of the wavelet cryptosystem

The wavelet system is implemented using a two-band analysis-synthesis filter bank. The analysis and synthesis banks of a two-channel perfect reconstruction filter bank in which the synthesis filters $g_0(n)$ and $g_1(n)$ are the scaling sequence and mother wavelet of lengths N, respectively. More specifically, the analysis bank performs the wavelet transform and the synthesis bank performs the inverse wavelet transform. The blocks labelled $[\downarrow 2]$ downsample by a factor of two by taking every other sample, and those labelled $[\uparrow 2]$ increase the sampling rate by a factor of two by inserting one sample with value zero between each pair of samples in the input sequence. The sequences labelled $y_0(n), y_1(n)$ are the wavelet coefficients. The impulse responses of the digital filters must be related if the synthesis section is to invert the results of the analysis. In this chapter, we study block ciphers of length $N = 2M$.

Similar to the wavelet error control coding, the inverse wavelet transform is used for transmitting (encryption) the message and the wavelet transform is used for receiving (decryption). Figure 3.2 shows the elementary block that is used for the encryption. We refer to this block as an elementary encryption block. This elementary encryption block utilizes the inverse wavelet transform together with a demultiplexer that splits the input signal $x(n)$ into even index $x_0(n)$ and odd index $x_1(n)$ sequences. The symmetric property of the polyphase filters is exploited to enable the use of the polyphase representation [37, 38] of multi-rate filters to further simplify the structure of the encryption block into Figure 3.3. In this representation $e_{00}(n)$ is the even index polyphase components of the filter $g_0(n)$. This polyphase filter $e_{00}(n)$ can be any symmetric sequence of length M, where M is an odd

number that is determined by the key length in the cryptosystem. This simplification also reduces the number of operations by a factor of four because the length of the polyphase filter $e_{00}(n)$ is half of the length of the filter $g_0(n)$. In summary, the elementary encryption block maps the sequence $x(n)$ by a one-to-one mapping to the sequence $y(n)$.

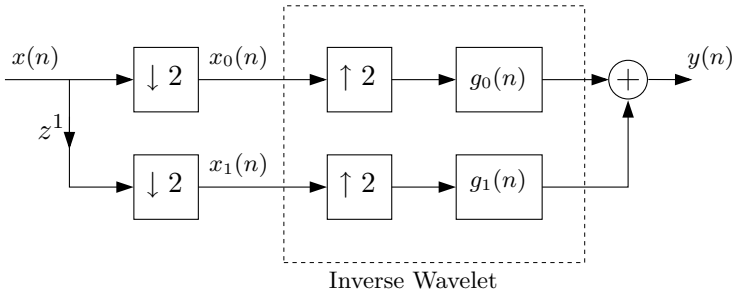


Figure 3.2: Multirate filter implementation of the elementary encryption block (EB).

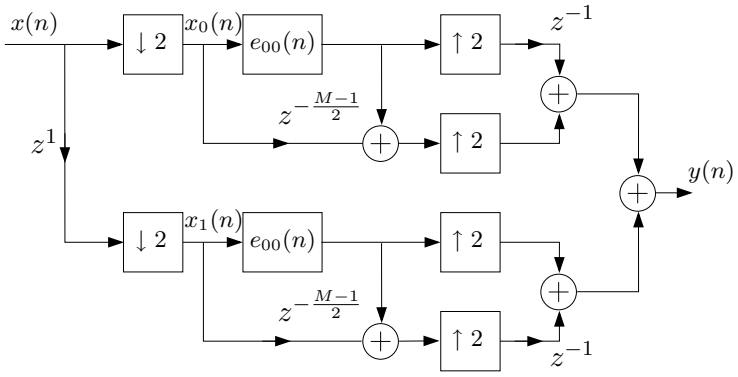


Figure 3.3: Polyphase representation of the encryption block in fields of characteristic two.

The mapping that is performed by the encryption block (EB) is a linear and bijective transformation. By the property of the wavelet system, one can easily show that $x(n)$ can be extracted from $y(n)$ by using the inverse system that is shown in Figure 3.4. We refer to this inverse system as a decryption block. The decryption block consists of the wavelet transform associated with the inverse wavelet transform in the encryption block and a multiplexer that interleaves and combines the even and odd indexes to get $x(n)$. Similar to the encryption block, the decryption block defined over the field $\mathbb{GF}(2^r)$ can be simplified to Figure 3.5 if it is represented by the symmetric polyphase component $e_{00}(n)$

of the filter $g_0(n)$.

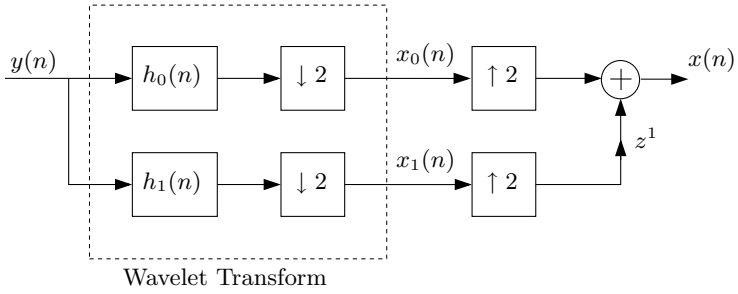


Figure 3.4: Multirate filter implementation of the elementary decryption block (DB).

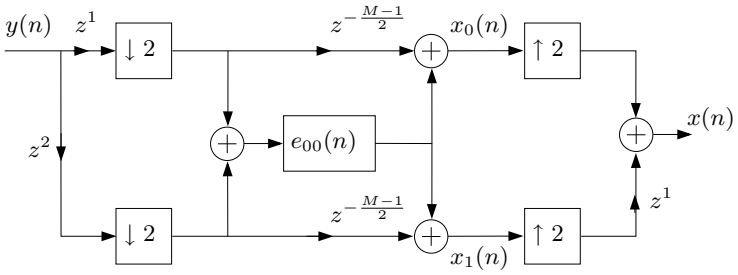


Figure 3.5: Polyphase representation of the decryption block in fields of characteristic two.

Figures 3.2 and 3.4 represent the basic blocks for encryption and decryption, respectively. Therefore, to determine both the encryption and decryption blocks, it is sufficient that we pick any arbitrary symmetric sequence $e_{00}(n)$ of length M, where M is any odd number. The output y of the wavelet transform can be represented with matrix operations as a function of $x(n)$ and the secret coefficients, $e_{00}(n)$. These derivations are detailed in Appendix A.

Using (2.4), it can be shown from Figure 3.2 that the input and output relation of the elementary encryption block can be written as

$$y = (G_0A + G_1B)x, \tag{3.1}$$

where the input and output are represented by two vectors x and y, respectively. Using the polyphase representation of the wavelet transform, the input and output relation of the elementary encryption block in Figure 3.3 can be written by

$$Y = (DE_{00}A + CE_{00}B + C(E_{00} + S)A + D(E_{00} + S)B)x \tag{3.2}$$

$$Y = Tx. \tag{3.3}$$

We represent the elementary encryption block by the matrix T. Similar matrix representation exists for the decryption block of Figure 3.5. We represent the decryption block by the matrix, F, for the remainder of this chapter. The matrices A, B, C, D, E, and S used in (3.1) and (3.2) are defined in Appendix A.

3.2.2 Nonlinear blocks of the wavelet cryptosystem

This section describes how the nonlinearity is added into the wavelet transform. Since the original wavelet system is linear, it is necessary to construct a nonlinear wavelet to make the system resistant against cryptanalytic attacks. This can be done by the lifting method. As shown in Figure 3.6, the nonlinearity in the encryption block is introduced by a implementing a feedback system that takes the output block y of the wavelet system, and after delaying by one block, it passes through a nonlinear operation and adds the result to the incoming message block (plaintext) x. The nonlinear operation is a mapping of every nonzero element of the block vector $y(n-N)$ (note that the delay is a block delay) to its inverse in $\mathbb{GF}(256)$. If the block vector $y(n-N)$ contains a zero element, we simply map the element to zero because a zero element does not have any inverse in $\mathbb{GF}(256)$. One can easily show that the nonlinear decryption block associated with the nonlinear encryption block is given by a feed-forward system as shown in Figure 3.7.

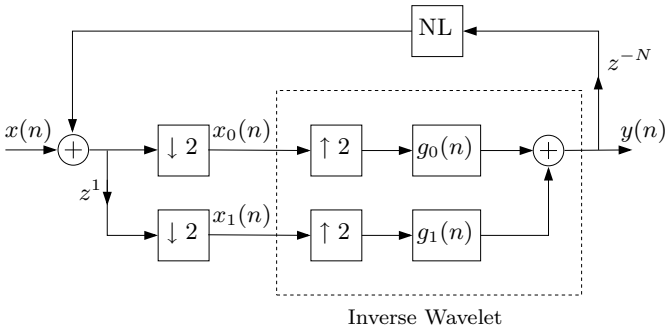


Figure 3.6: Elementary nonlinear encryption block (NLEB).

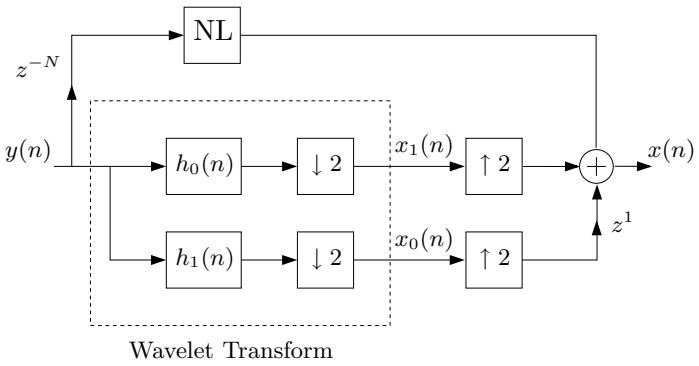


Figure 3.7: Elementary nonlinear decryption block (NLDB).

3.3 Two-round wavelet cryptosystem

3.3.1 General structure of WBC

This section presents the overall proposed block cipher variation of the wavelet cryptosystem, the wavelet block cipher (WBC). As shown in Figures 3.8 and 3.9, the wavelet encryption system consists of two rounds. As we will realize later, two rounds will result in the total key size of 128 bits. Based on the application, the number of rounds can be increased to reach a higher security level. These rounds are identical except for that the filters in each round are different from one another. Each round consists of two elementary blocks. The first one is the nonlinear encryption block (NLEB) that is implemented by the polyphase representation form that we discussed earlier. The second block is the linear encryption block (EB), which is also realized by the polyphase representation. Each elementary block consists of a polyphase filter of length $M = 15$ that operates over the finite-field $\mathbb{GF}(256)$.

As we discussed earlier, the polyphase filter coefficients are the secret key (unknown for the adversary) in the wavelet cryptosystem. Since the polyphase filter is symmetric, the actual length of the key is eight coefficients in $\mathbb{GF}(256)$, or equivalently 64 bits. Therefore, each round of the wavelet encryption system has a key size of 128 bits. Note that the effective key size of the two-round wavelet is still 128 bits. This is because the key for the second round is obtained by the same 128-bit key of the first round by a (publicly known) bitwise permutation of the coefficients. Figure 3.9 shows the wavelet decryption system

which consists of two rounds as in the encryption system. Each round inverts the operation of the corresponding round of the encryption system. Similar to the encryption blocks of the wavelet encryption system, the decryption blocks of the wavelet decryption system are implemented by the polyphase form that we have already explained. Based on the filter length that is used, in the block cipher case, the length of the message block is $N = 30$ symbols in $\mathbb{GF}(256)$, or equivalently 240 bits. Notice that the number of operations that are required by the convolution of the encryption and decryption process can be significantly reduced by using finite-field FFT or an appropriate bilinear cyclic convolution transform that was developed in [74].

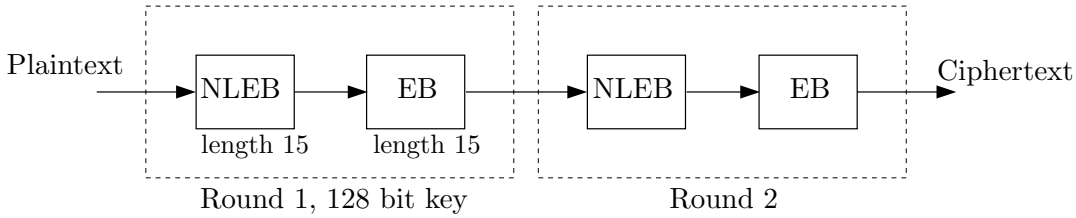


Figure 3.8: The wavelet encryption system

Since the input is processed block-by-block in the block cipher mode, the nonlinear part of the wavelet cryptosystem that contains feedback also operates in a vector (block) by vector (block) form. This implies that we buffer the output of the feedback system for every block of 30 symbols. The block of the feedback output will be added to the next message block that is going to be encrypted. In other words, the present feedback output does not affect the encryption of the current message. Instead, the current message is added to the feedback output (240 bits) resulting from the encryption of the previous message block.

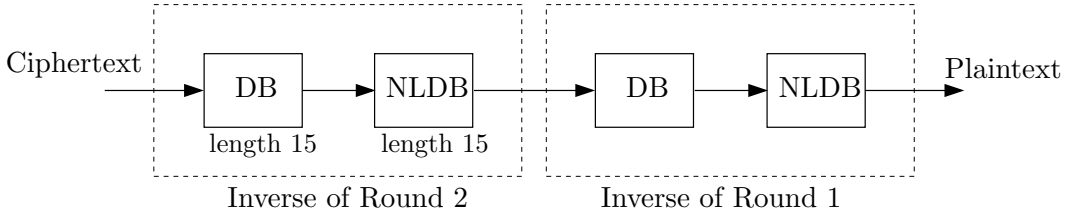


Figure 3.9: The wavelet decryption system.

Every time the system starts the encryption process, the polyphase filters in the encryption blocks are started from the zero initial states. For higher security, we may assume that a random initial sequence of the input block size is added to the plaintext whenever the system starts the encryption. Since we do not want to transmit this random vector to the receiver, the first block of the encrypted message will not be recoverable by the receiver and it should be disregarded. However, the subsequent blocks will be correctly recovered by the receiver without requiring a knowledge about the random initial sequence.

In Section 3.2.1, the polyphase representation of the elementary encryption and decryption blocks is shown in Figures 3.3 and 3.5. The encryption has two convolution operations, while the decryption only has one. Because of this structure in the proposed wavelet cryptosystem, the decryption is almost two times faster (and less complex) than the encryption. The polyphase representation of the elementary encryption and decryption blocks is shown in Figures 3.3 and 3.5. The encryption has one more convolution operation, which is the most computationally intensive operation in the algorithm, than the decryption.

In this way, the decryption for the wavelet cryptosystem is almost two times faster (and less complex) than the encryption, so exchanging the role of the transmitter and receiver may fit some applications, where the computational load is higher in the transmitter. This can be done by exchanging the role of the wavelet transform (analysis bank) and the inverse wavelet (synthesis bank). In other words, as shown in Figures 3.10 and 3.11, the wavelet transform is used in the elementary encryption block and the inverse wavelet transform is used for the basic decryption block. Figures 3.10 and 3.11 shows the structure of the nonlinear encryption and decryption blocks when we exchange the roles of the wavelet and the inverse wavelet. A similar change can be applied for the linear encryption and decryption blocks used in the wavelet cryptosystem shown in Figures 3.8 and 3.9.

3.3.2 Key generation for WBC

This section describes the key exchange between two parties, say Alice and Bob, so that they end up establishing the same filters in a secure manner. We are going to describe two

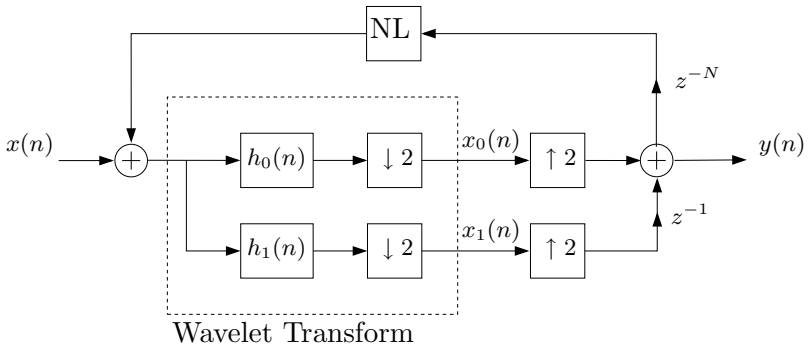


Figure 3.10: Nonlinear transform block constructed by the wavelet transform (NLEB).

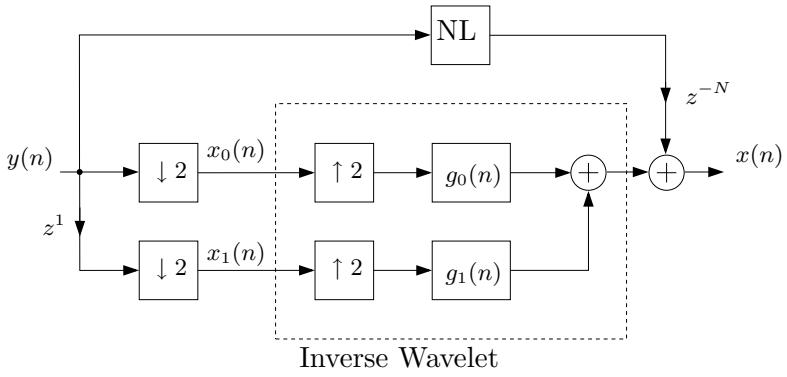


Figure 3.11: Nonlinear inverse transform block constructed by the inverse wavelet transform (NLDB).

different methods based on the two methods of the filter generations. Note that we only need to generate the 128 bit key (16 symbols in $\mathbb{GF}(256)$) of the first round. The coefficients of the second round are obtained by the permutation of these bits. This permutation is public knowledge. As an example, the Diffie-Hellman key exchange [28] protocol can be used to setup the filter coefficients, yet it is possible to employ any public key exchange protocol. In an encryption system using the Diffie-Hellman key exchange, Alice (user A) sends an invoice to Bob (user B), encrypting it via her secret key and user Bob's public key. Bob then uses his private key and Alice's public key to decrypt the transmitted document. In the symmetric case, only $(M+1)/2$ key symbols must be exchanged, whereas for the nonsymmetric case, M exchanges must be done.

3.3.2.1 Symmetric Polyphase Filters

In the two-round wavelet cryptosystem, the key exchange protocol must ensure the same polyphase filters $e_{oo}^{(1)}(n)$ and $e_{oo}^{(2)}(n)$ for both encryption and decryption. Here, the filters $e_{oo}^{(1)}(n)$ and $e_{oo}^{(2)}(n)$ are the even index polyphase components of the nonlinear encryption block and the linear encryption block of the first round, respectively. Therefore, two symmetric filters $e_{oo}^{(1)}(n)$ and $e_{oo}^{(2)}(n)$ are securely generated by a method relying on the security of the discrete log problem (DLP) over finite-fields.

Alice and Bob each independently choose a symmetric sequence of length M as their secret key, where M is an odd positive integer. In the symmetric case, $(M+1)/2$ key symbols will be exchanged. Let the symmetric sequence $\{k_0, \ldots, k_{M-1}\}$ be the secret key of Alice. Also let the symmetric sequence $\{v_0, \ldots, v_{M-1}\}$ be the secret key of Bob. If the Diffie-Hellman key exchange is used, the resulting shared key will be $\{\alpha_0^{k_0 v_0}, \ldots, \alpha_{M-1}^{k_{M-1} v_{M-1}}\}$, using the discrete log function over a cyclic group $\mathbb{Z}_p^*$, where p is a large prime and such that the DLP is intractable over $\mathbb{Z}_p^*$, and α is the generator of $\mathbb{Z}_p^*$.

3.3.2.2 Nonsymmetric polyphase filters

The filter banks used in WBC employ symmetric polyphase filters. This symmetric property is useful to simplify the polyphase implementation of the filter banks and reduces the number of additions and multiplications that are required by the wavelet cryptosystem.

However, the symmetric structure of the polyphase filters may reduce the strength of the security for the wavelet cryptosystem. Therefore, as an alternative to the symmetric method, we introduce a second method that eliminate this concern at the cost of quadruple the computational complexity in the decryption block (the computational complexity is doubled for the encryption block). It is very important to note that the polyphase representation of the two-band filter banks as shown in Figures 3.3 and 3.5 is valid only if symmetric polyphase filters are used. The general well-known polyphase representation (that is introduced for real/complex fields) should be used with the nonsymmetric method. It is known that every filter bank has a polyphase matrix representation. Two types of building blocks, $D(z)$ and $S(z)$, are necessary and sufficient building blocks to construct all the two-band filter banks [40]. In the following, these building blocks are described.

$\mathbf{D}_v(z)$ is the degree-one PU building block defined by:

$$\mathbf{D}(z) = d(0) + z^{-1}d(1) = I + \ell_v^{-1}vv^T + z^{-1}\ell_v^{-1}vv^T, \text{ where : } \ \ell_v = v^Tv \neq 0. \tag{3.4}$$

Note that for two-band filter banks $v = [a\ b]^T$ is a vector of length two in $GF(2^r)$ and ℓ_v is always square. The nonzero condition of ℓ_v requires that $a \neq b$.

A degree 2τ elementary building block has the following structure:

$$\mathbf{S}_{\tau,\zeta}(z) = \zeta(I + J) + z^{-\tau}I + z^{-2\tau}\zeta(I + J), \tag{3.5}$$

where $\zeta \neq 0$ is a scalar in $GF(2^r)$ and τ is any positive integer. Here, I and J are the identity and exchange matrices, respectively. Thus, $I + J = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$.

To generate 16 symbols key that are shared by the transmitter and receiver, again the key exchange protocol using the discrete log problem (DLP) can be used. The generation the polyphase filters once the 16 shared key coefficients are established is described. Since the first round consists of two sets of two-band filter banks, thus each filter banks uses eight key symbols to specify their filter coefficients. This implies that the polyphase matrix should have the following form $E(z) = D_{v_1}(z)S_{2,\zeta_1}(z)D_{v_2}(z)S_{2,\zeta_2}(z)D_{v_3}(z)$ where the scalars ζ_1, ζ_2 and the vectors v_1, v_2, v_3 satisfy the criterion that was described previously. Each vector v is specified by two key coefficients and each ζ is determined by one key coefficient.

Therefore, the total key usage to specify the polyphase matrix is 8. By this description, the lengths of the polyphase filters are equal to 8 as opposed to the symmetric polyphase method for which we need length 15 polyphase filters for each two-band filter banks. Since the length 15 convolution can be performed more efficiently than length 8 convolution in GF(256), we may increase the length of the polyphase filters to 15. This can be done by using for example $E(z) = D_{v_1}(z^2)S_{2,\zeta_1}(z^2)D_{v_2}(z^2)S_{2,\zeta_2}(z^2)D_{v_3}(z^2)$. However, the security of the system benefits from the use of $E(z) = D_{v_1}(z)\left[\prod_{i=1}^{4} S_{2,\zeta_i}(z)D_{v_{i+1}}\right]D_{v_6}(z)$. This way, the resulting polyphase filter has length 15, but the key size is doubled. In other words, with the same computational cost of the encryption and decryption, the key size becomes 256 instead of 128 bits.

3.3.2.3 Key schedule

As discussed, the filter coefficients of the first round of the wavelet cryptosystem, $e_{00}^{(1)}$ and $e_{00}^{(2)}$ are defined by using the 16 key symbols. The filter coefficients of the second round, $e_{00}^{(3)}$ and $e_{00}^{(4)}$, are obtained by some permutation on the original 128 key bits. Our cryptanalysis in Section 3.4.5 suggests that to prevent the DFT attack, a bitwise permutation within a symbol of $e_{00}^{(i)}(n)$ should be used to obtain the corresponding symbol in $e_{00}^{(i+2)}$ for $i = 1, 2$. In other words, the j^{th} coefficient of $e_{00}^{(3)}(n)$ is constructed by eight bit permutations of the j^{th} coefficient of $e_{00}^{(1)}(n)$. Therefore, only one 8-bit permutation is necessary for the system.

As a remark, this chapter only considers the two-band wavelet (i.e., two-channel filter bank) over finite fields to construct the wavelet cryptosystem. Multi-band wavelets (i.e., multi-channel filter banks) are the generalization of the two-band wavelets. The analysis and synthesis of a multi-channel filter bank is illustrated in Figure 3.12. The theory that describes how to construct multi-channel filter banks over fields of characteristic two is presented in [39, 40, 37]. The wavelet cryptosystem can also be constructed by multi-band wavelets (i.e., multi-channel filter banks). To explain this, consider the nonlinear transform and inverse transform blocks shown by Figure 3.6. We replace the inverse wavelet (synthesis bank) of the two-band system shown in Figure 3.6 with the inverse wavelet (synthesis bank) of the L-band system shown in Figure 3.12. Similarly, the wavelet transform (analysis bank)

of the two-band system shown in Figure 3.7 can be replaced with the wavelet transform (analysis bank) of the L-band system. Additionally, we split the message sequence $x(n)$ into L subsequences as opposed to only two subsequences in Figures 3.6 and 3.7. In other words, all the upsamplers and downsamplers by factor of two are replaced by the factor L. These modifications are required to construct the elementary linear encryption and decryption blocks in order to use L-channel filter banks instead of the two-channel filter banks.

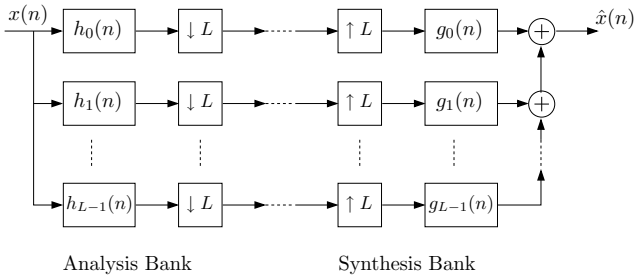


Figure 3.12: L-channel maximally decimated filter bank

3.4 Cryptanalysis of WBC

We now consider the strength of security with regard to known cryptanalytic techniques and also present some new cryptanalytic methods specific to the wavelet structure. We propose some attacks on the reduced round wavelet cryptosystem as well as the best-known attack on the full 2-round system. Currently, we have found no realizable attacks that threaten the security of the wavelet cryptosystem. It is worthwhile to note that the structure of the one-round encryption is similar to the cascade of cipher-block chaining and electronic codebook (CBC|ECB) modes of operation. Therefore, their cryptanalytic methods constructed in [4, 5] may present potential cryptanalytic methods for the wavelet algorithm. The difference in the structure comes with the addition of the Inv function in the feedback branch. Also, we note that the cryptographic primitive of wavelet cryptosystem has an inherent feedback structure, much like the CBC mode of operation. Therefore, it is viable to use the wavelet cryptosystem in any mode of operation, including ECB. However, in the encryption using ECB (which in this case it actually has feedback), the first block is discarded and cannot be decrypted because of the use of the random initial vector. This does not affect the

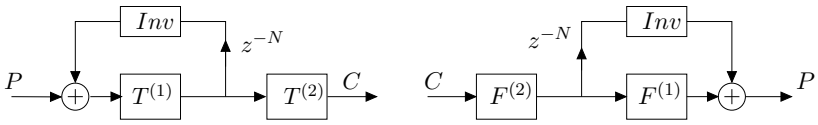
analysis of the cryptanalytic attacks. This section first considers properties of WBC and an approach to analyze WBC security. Second, we examine the vulnerabilities of WBC to linear and differential attacks. Third, we consider algebraic attacks to take advantage of its highly mathematical structure.

The wavelet decryption structure is illustrated in Figures 3.13 and 3.14 for one and two rounds. These figures are simply the explicit declaration of Figure 3.8, where the function NL is Inv and the $T^{(i)}$ and $F^{(i)}$ are matrix representations of the EB and DB blocks, respectively. For the wavelet encryption, the feedback as shown in Figure 3.14 results in a system that every ciphertext block is determined by the current plaintext block and each previously encrypted ciphertext block. This impedes the ability to analyze the system since every plaintext block after the system's initial conditions must be considered. Therefore, we choose to consider the wavelet decryption, or a chosen-ciphertext attack (CCA) for which the system is feedforward. Figures 3.15 and 3.16 show the unwrapped versions of Figures 3.13 and 3.14, respectively. Each decrypted plaintext block is dependent on the current ciphertext block and the two previous ciphertext blocks. All attempts at attacks on the wavelet cryptosystem presented in subsequent sections will be CCA and according to Figures 3.13 and 3.14.

For wavelet encryption, full diffusion is achieved rapidly as a result of the algorithm's structure, which holds a high dependence on the key. The effect of an input XOR difference propagates quickly to each bit of the output. If one elementary wavelet encryption block (Figure 3.2) is considered, the relationship between the input x and output y is given by $y = Tx$, recalling that T contains all the key information, then every bit of C is dependent on every value of P. Hence, the full diffusion property is satisfied, within one elementary encryption block.

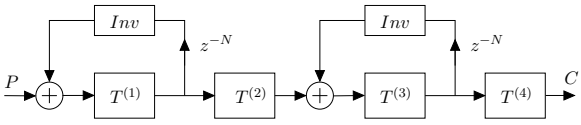
It must be noted that keys with many elements equal to zero do not possess such properties and should not be used. The weak key is noticed by observing Figure 3.16 and the delta attack described in Section 3.4.4. First, recall that $F^{(2)}$ and $F^{(4)}$ are generated by permutations of the same key, $e_{oo}^{(2)}(n)$. By tracking the $C^{(M-2)}$ branch, it can be shown that the minimum number of nonzero elements of $e_{oo}^{(2)}(n)$ must be five. This protects against

having this branch to not be fully diffused. Furthermore, by observing the $C^{(M)}$ branch along with the requirements for $e_{oo}^{(2)}(n)$, it also can be shown that the minimum number of nonzero elements of $e_{oo}^{(1)}(n)$ (which generate $F^{(1)}$ and $F^{(3)}$) is one. Therefore, ciphertexts using keys not satisfying these conditions may not maintain the full diffusion property. This amounts to one weak key out of every 7.2×10^{22} keys chosen. Keys satisfying these conditions but still with some key symbols equal to zero may appear to be susceptible to attacks shown in subsequent sections; however, no method has been found yet to exploit this concept of taking advantage of the key not containing all nonzero elements.

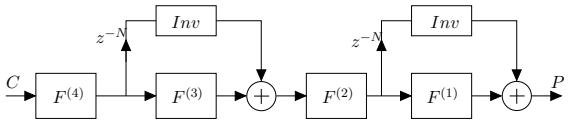


(a) One-round wavelet encryption. (b) One-round wavelet decryption.

Figure 3.13: The one-round wavelet system



(a) Two-round wavelet encryption.



(b) Two-round wavelet decryption.

Figure 3.14: The two-round wavelet system

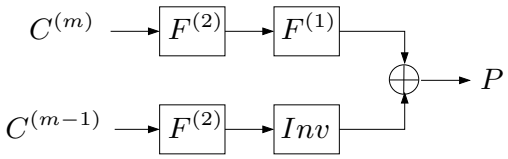


Figure 3.15: Unwrapped decryption structure for one round

3.4.1 Resistance to linear and differential cryptanalysis

Linear and Differential attacks [6, 67] are two classical cryptanalytic attacks that have exposed security vulnerabilities in many cryptographic algorithms. Therefore, it is necessary

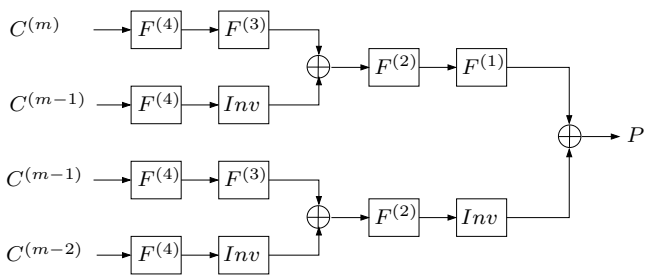


Figure 3.16: Unwrapped decryption structure for two rounds

to make sure that the wavelet algorithm is not susceptible to such attacks. The viability of these attacks is based on the presence of strong characteristics. Occurring with certain probability, a characteristic is a relationship of plaintext, ciphertext, and key bits that span one or more rounds of the elementary round function. Attacks are formed by concatenating these characteristics to establish a relationship between the plaintext, ciphertext, and key bits of the full cryptographic algorithm. The linear and differential characteristics were defined in Section 2.2.1.

Attacks (perhaps of reduced-round versions of an algorithm) are mounted by identifying differentials that are only affected by a small subset of the blocks of plaintext, ciphertext and key. For example, differential characteristics in DES are found such that only few of the 8 S-boxes (per round) are involved. Indeed, there have been some sophisticated improvements (last-round trick, symmetric characteristics) which have been shown to reduce the complexity of these attacks, but our justification considers the lack of the basic characteristic needed of both the linear and differential attacks.

For the differential characteristic, it is important to note that the elementary wavelet encryption block and the finite-field inverse nonlinear block are bijective mappings. Consider the bijective function F, which is the elementary decryption block. For a differential characteristic, we consider C, C' for a fixed C^* and observe the distribution of $P \oplus P' = FC \oplus FC'$. It is easy to see that every value of $P \oplus P'$ will occur 2^N times; therefore every characteristic for one block will have probability 2^{-N}. Furthermore, every differential will be satisfied with this probability. So, the differential attack using XOR as the differential is not possible.

For a linear characteristic, consider an elementary wavelet decryption block, $x = Fy$, and

assume that there exists some correlation between symbols of a subset of bits of the plaintext, ciphertext and key, namely $\{P_{i_1} \cdots P_{i_m}\}, \{C_{j_1} \cdots C_{j_n}\}, \{K_{r_1} \cdots K_{r_l}\}$. The function F can be represented by a matrix where each F_{ij} depends on the key symbols. Therefore, according to the definition of the characteristic,

$$P_{i_1} + \cdots + P_{i_m} = C_{j_1} + \cdots + C_{j_n} + K_{r_1} + \cdots + K_{r_l} \tag{3.6}$$

we have,

$$F_{11}C_1 + \cdots + F_{1N}C_N + \cdots + F_{i_m1}C_1 + \cdots + F_{i_mN}C_N = C_1 + \cdots + C_n + K_1 + \cdots + K_l \tag{3.7}$$

For an arbitrary choice of key, which in this case means the arbitrary selection of all F_{ij} and K_i, (3.7) and the full diffusion property discourages the possibility for the characteristic to hold with high probability. We have not found any characteristics that potentially challenge the security of the algorithm in this fashion. This demonstrates the high key dependency and rapid rate of full diffusion thats leads to a lack of linear characteristics for the elementary wavelet blocks. Because of this property, strong linear and differential characteristics that pose any serious threats to the wavelet algorithm do not exist. In the following sections, we try to exploit the feed-forward structure in the decryption of the two-round wavelet cryptosystem to find new types of attacks.

3.4.2 Divide-and-conquer linear attack on the one-round WBC

This section presents chosen ciphertext attacks on the wavelet algorithm. By unwrapping the feed-forward structure of the wavelet decryption as in Figure 3.15 and 3.16, we will consider sets of ciphertext blocks to produce a single decrypted plaintext block. Denote this set $\mathbf{C} = [\ C^{(m)}\ C^{(m-1)} \dots\ ]$. Here, an attack considers ciphertexts in sets of two for one-round attacks and three for the two-round attacks. Perhaps there exist more fortuitous choices of these ciphertexts, but for now, this approach will suffice.

First of all, it is obvious that with N linearly independent observations of the input and their corresponding output of a linear elementary decryption block, its individual key values can be determined. So, attempting to exploit the wavelet decryption structure to

reduce the problem of analyzing nonlinear wavelet decryption blocks to analyzing the linear variety is the first approach taken.

The key for the one-round wavelet system can be recovered by choosing $\mathbf{C} = [\ C^{(m)}\ C^{(m-1)}\ ] = [\ 0\ \mathbf{x}\ ]$, where $\mathbf{x}$ is any non-zero vector of length N. The key bits within $F^{(2)}$ are obtained choosing $\mathbf{C} = [\ 0\ \ \mathbf{x}\ ]$ to obtain $Inv(\mathbf{P}) = F^{(2)}\mathbf{x}$. Therefore, by choosing N linearly independent ciphertext blocks, $\mathbf{X} = [\mathbf{x_1} \cdots \mathbf{x_n}]$, we can obtain $F^{(2)}$ from $F^{(2)} = Inv(\mathbf{P})\mathbf{X}^{-1}$.

Now, $F^{(1)}$ is determined by choosing $\mathbf{C} = [\ \mathbf{x}\ 0\ ]$ to obtain the relation $\mathbf{P} = F^{(2)}(F^{(1)}(\mathbf{x}))$. Therefore $F^{(1)}$ can also be obtained by choosing N linearly independent ciphertext blocks, $\mathbf{x}$. Thus, this attack requires at most $2N$ chosen ciphertexts.

Note that a similar CCA technique is not applicable to the two-round wavelet cryptosystem due to the second feedforward branch and associated nonlinear device in the decryption structure which define the output from the input. The next several sections are devoted to attempts at other approaches to find an attack with a lower computational complexity than the previously stated method.

3.4.3 Analysis of the interpolation attack

In this section, we describe a variation of the interpolation attack [49] that exploits the algebraic structure of the wavelet cryptosystem. The approach of this chosen ciphertext attack is to characterize the plaintext as a function of a rational polynomial of the ciphertext symbols and solve these as a linear system of equations. This provides the attacker with the ability of ciphertext decryption; however, the attacker does not recover the key.

The order of complexity is determined by the number of coefficients in the polynomial. Also observe that the wavelet algorithm has a key space of 128 bits. Therefore, an attack is effective if its complexity is less than $O(2^{128})$. Throughout the interpolation attack section, we will consider one index (without loss of generality, the first index) of the plaintext for clarity. Therefore, the total complexity of the computation for the entire plaintext block is increased by a factor of N relative to the number of coefficients that characterize a single index.

One index of the matrix-vector multiplication for an elementary wavelet linear wavelet

block is represented by $P = \sum_{k=1}^{N} F_{1k} C_k$. For one elementary decryption block, each plaintext symbol is represented by a polynomial with N terms. Thus, N linearly independent ciphertexts are required to completely determine the system. This is the basis for the remainder of the sections that consider the interpolation attack. We also abuse the notation for $\frac{1}{\mathbf{x}}$. By this, we mean $Inv(\mathbf{x})$, which is the index by index mapping of the vector $\mathbf{x}$ to its inverse element in $\mathbb{GF}(256)$.

We establish this fact for use when considering rational expressions of polynomials. Let L and M be two elementary decryption blocks and C be the ciphertext block. This fact counts the number of coefficients of $P = L(Inv(MC))$.

Fact 1. *$L(\frac{1}{MC})$ results in a polynomial with N^{N+1} unknown terms.*

Proof.

$$L\begin{bmatrix} \frac{1}{\sum_{k=1}^{N} M_{1k}C_k} \\ \vdots \\ \frac{1}{\sum_{k=1}^{N} M_{Nk}C_k} \end{bmatrix} = \begin{bmatrix} \frac{L_{11}}{\sum_{k=1}^{N} M_{1k}C_k} + \cdots + \frac{L_{1N}}{\sum_{k=1}^{N} M_{Nk}C_k} \\ \vdots \\ \frac{L_{N1}}{\sum_{k=1}^{N} M_{1k}C_k} + \cdots + \frac{L_{NN}}{\sum_{k=1}^{N} M_{Nk}C_k} \end{bmatrix} = \begin{bmatrix} \frac{\sum_{h=1}^{N} L_{1h} \prod_{j\neq h}^{N} \sum_{i=1}^{N} M_{ji}C_i}{\prod_{k=1}^{N} \sum_{i=1}^{N} M_{ji}C_i} \\ \vdots \\ \frac{\sum_{h=1}^{N} L_{Nh} \prod_{j\neq h}^{N} \sum_{i=1}^{N} M_{ji}C_i}{\prod_{k=1}^{N} \sum_{i=1}^{N} M_{ji}C_i} \end{bmatrix} \tag{3.8}$$

Altogether, there are N^N coefficients in both the numerator and the denominator, so there are N^{N+1} unknowns in total to characterize P. □

3.4.3.1 Interpolation Attack of the One-Round Wavelet Decryption

This section presents the attacks for the wavelet algorithm; in this case, the one round system. Using Figure 3.15, the attacker can characterize one round of the wavelet decryption by

$$P = F^{(12)}C^{(m)} + \frac{1}{F^{(2)}C^{(m-1)}} \tag{3.9}$$

Therefore, one symbol of the plaintext is given by

$$\left[P_1^m\right] = \left\lfloor \frac{\sum_{k=1}^{N} \mathbf{F}_{1k}^{(12)} C_k^{(m)} \sum_{j=1}^{N} \mathbf{F}_{1j}^{(1)} C_j^{(m-1)} + 1}{\sum_{k=1}^{N} \mathbf{F}_{1k}^{(1)} C_k^{(m-1)}} \right\rfloor \tag{3.10}$$

As shown in (3.10), the rational expression of polynomials in terms of the ciphertext consists of N^2 unknown coefficients in the numerator and N unknowns in the denominator. For each index of the plaintext, $N^2 + N$ known ciphertexts are required to characterize the unknown coefficients. Therefore, the total complexity of ciphertext decryption attack on one round requires solving $\mathrm{O}(N(N^2 + N)) = \mathrm{O}(2^{14.8})$ linear equations (Note: $N = 2M = 30$).

The complexity of the attack can be improved by employing a divide and conquer strategy. Since each of the branches in in Figure 3.15 can be analyzed independently, the attacker can choose the input to one of the branches to be zero thus allowing the coefficients of the other branch to be identified separately from the other input. The divide and conquer variation of the interpolation attack on one round uses chosen ciphertexts $C^{(m)}$ and $C^{(m-1)}$, where each term has N unknowns. Each index of the plaintext can be solved with $2N$ chosen ciphertexts. Therefore, the attack requires solving $\mathrm{O}(2N^2) = \mathrm{O}(2^{10.8})$ linear equations. Note that the complexity of this attack is 16 times faster than the initial interpolation attack, which has a complexity of $\mathrm{O}(2^{14.8})$. Nonetheless, both of these attacks are far below the exhaustive key search complexity of $\mathrm{O}(2^{128})$. In this case, the attacker is able to gain the ability of ciphertext decryption, but has not recovered information about the key.

3.4.3.2 Security of the two-round wavelet cryptosystem against the interpolation attack

In this section, we pursue the same interpolation attack for the two round wavelet encryption. Since the divide and conquer method aided in reducing the complexity of the one-round attack, we will use this variety of the attack for the two-round case. The two-round wavelet decryption is described by

$$P = F^{(1234)} C^{(m)} + F^{(34)} \left(\frac{1}{F^{(1)} C^{(m-1)}} \right) + \frac{1}{F^{(123)} C^{(m-1)}} + \frac{1}{F^{(3)} \left(\frac{1}{F^{(1)} C^{(m-2)}} \right)} \tag{3.11}$$

which is divided into three independent expressions in terms of the three ciphertexts, $C^{(m)}$, $C^{(m-1)}$, $C^{(m-2)}$, on which the plaintext is dependent. This is illustrated in Figure 3.16. We can separately consider the complexity of each polynomial now. For the first term,

$$\left[P_1^m\right] = \left[\sum_{k=1}^{N} \mathbf{F}_{1k}^{(1234)} C_k^{(m)}\right] \tag{3.12}$$

the polynomial dependent on $C^{(m)}$ is simply $F^{(1234)}$. This polynomial has N unknowns (the product of the four wavelet transformations). Therefore, it can be solved with N^2 chosen ciphertexts of $C^{(m)}$.

The second term of the expression for $C^{(m-1)}$ is described by

$$P = F^{(34)}\left(\frac{1}{F^{(1)}C^{(m-1)}}\right) + \frac{1}{F^{(123)}C^{(m-1)}} \tag{3.13}$$

We can write:

$$P_1 = \sum_{k=1}^{N} \mathbf{F}_{1k}^{(34)} \left[\frac{1}{\sum_{i=1}^{N} \mathbf{F}_{ki}^{(1)} C_i^{(m-1)}}\right] + \left[\frac{1}{\sum_{j=1}^{N} \mathbf{F}_{1j}^{(123)} C_j^{(m-1)}}\right] \tag{3.14}$$

Using Fact 1, the first term becomes

$$P_1 = \frac{\sum_{k=1}^{N} \mathbf{F}_{1k}^{(123)} C_k^{(m-1)} \sum_{h=1}^{N} \mathbf{F}_{1h}^{(34)} \prod_{j=1, j\neq h}^{N} \sum_{i=1}^{N} \mathbf{F}_{ji}^{(1)} C_i^{(m-1)} + \sum_{h=1}^{N} \mathbf{F}_{1h}^{(34)} \prod_{j=1, j\neq h}^{N} \sum_{i=1}^{N} \mathbf{F}_{ji}^{(1)} C_i^{(m-1)}}{\sum_{k=1}^{N} \mathbf{F}_{1k}^{(123)} C_k^{(m-1)} \prod_{k=1}^{N} \sum_{i=1}^{N} \mathbf{F}_{ji}^{(1)} C_i^{(m-1)}} \tag{3.15}$$

Equation (3.15) is used to determine the number of unknown coefficients of the polynomial in C for one index of P. The denominator consists of N^{N+1} terms; these are all of the N+1 degree terms that are comprised of every combination of the indices of C, $\{C_1, C_2, \ldots C_N\}$. Likewise, the numerator is comprised of all combinations of terms of degree N and all terms of degree N+1. Therefore, the complexity of (3.13) is $N(2N^{N+1} + N^N)$

The third term is the path that traverses across two Inv functions. The $C^{(m-2)}$ term is described by

$$P_1 = \frac{1}{F^{(3)}(\frac{1}{F^{(1)}C^{(m-2)}})} \tag{3.16}$$

Using Fact 1, the third term of the two-round interpolation attack is described by

$$P_1 = \frac{\prod_{j=1}^{N} \sum_{k=1}^{N} \mathbf{F}_{jk}^{(1)} C_k^{(m-2)}}{\sum_{h=1}^{N} \mathbf{F}_{1h}^{(3)} \prod_{i=1, i\neq h}^{N} \sum_{k=1}^{N} \mathbf{F}_{ik}^{(1)} C_k^{(m-2)}} \tag{3.17}$$

The third term will have $N(2N^N)$ unknown coefficients for the entire plaintext. In total, the two round divide-and-conquer variation of the interpolation attack has a complexity of $\mathrm{O}(N(2N^N + 2N^{N+1} + N^{N+1} + N)) = O(2^{158})$ for $N = 30$. This is still of higher complexity than the exhaustive key search for the wavelet cryptosystem, $\mathrm{O}(2^{128})$.

3.4.4 Analysis of the delta function attack

This section considers another approach, which is a chosen ciphertext attack where the ciphertexts are delta functions. The motivation of this attack is that, with a minimal input to the system, the polynomials involved in the interpolation attack would have a lower complexity (i.e. less coefficients). According to Figure 3.16, an attack can consider one index of the three possible ciphertext blocks, $\{C^{(m)}, C^{(m-1)}, C^{(m-2)}\}$, to be a delta function input. This method attempts to analyze a delta input to the structure. This approach is similar to the divide and conquer variation of the interpolation attack; however, this method will attempt to recover the key, thus resulting in a stronger attack.

The complexity of this attack is dependent on the ability to solve systems of nonlinear equations. The extent of the analysis of this attack will be to provide an upper bound b for the complexity of these attacks. The complexity will be determined through the bound derived by Möller and Mora [34].

Fact 2. *Let*

$r =$ *the number of variables,*

$d =$ *the maximum degree of the polynomials g_i, the Grobner basis polynomials and*

$s =$ *the degree of the Hilbert polynomial (this is one less than the dimension; it is between 0 and $r - 1$)*

The bound is

$$b = ((r+1)(d+1)+1)^{2^{(s+1)}(r+1)}$$

To establish a lower bound on this sort of calculation, we will consider the case where $s = 0$. From Figure 3.16, we consider the two ciphertext blocks, $C^{(m)}$ and $C^{(m-2)}$. The $C^{(m-2)}$ contains key values from two wavelet transformations and the $C^{(m)}$ branch contains no Inv functions. By placing the delta function in the ciphertext $C^{(m)}$ (without loss of generality, the delta function in index 1, c_1), the resulting recovered plaintext is expressed by

$$P_i = C_1^{(m-2)} \sum_{h=1}^{N} \sum_{j=1}^{N} \sum_{l=1}^{N} F_{ih}^{(4)} F_{hj}^{(3)} F_{jl}^{(2)} F_{l1}^{(1)} \tag{3.18}$$

The delta function can also be placed in the ciphertext $C^{(m-2)}$; therefore, the recovered plaintext is expressed by

$$P_i = \left[\sum_{j=1}^{N} F_{ij}^{(3)} \left[F_{j1}^{(1)} C_1^{(m-2)} \right]^{-1} \right]^{-1}, \tag{3.19}$$

which can also be represented by

$$P_i = \frac{F_{11}^{(1)} F_{21}^{(1)} \cdots F_{N1}^{(1)} C_1^{(m-2)}}{\sum_{k=1}^{N} F_{ik}^{(3)} \prod_{j=1, j\neq k}^{N} F_{j1}^{(1)}}. \tag{3.20}$$

Equation (3.20) shows that P_i is a function of all of the elements of the $F^{(1)}$ and $F^{(3)}$ matrices. However, the two-round wavelet cryptosystem is only dependent on 16 key symbols. Therefore, P_i is a function of only 16 variables. Also, its maximum degree is N. According to (3.18), the complexity to solve this nonlinear system of equations is $O(2^{307})$. The calculated upper bound suggests that this attack will not be feasible for the wavelet cryptosystem. It is possible to reduce the complexity of the nonlinear systems of equations by relinearization explained in [21, 55]. These methods have not proved to offer any threat to the systems of equations generated with the wavelet encryption. Therefore, this offers some validity to the calculated upper bound of the system in response to the delta function attack.

3.4.5 Security against an attack using the discrete Fourier transform

Another approach that an attacker may take is to consider an alternative but identical representation of the wavelet cryptosystem. This section explores the potential for an attack using the discrete Fourier transform (DFT) to generate an alternative representation, which facilitates cryptanalysis of the wavelet algorithm.

3.4.5.1 Obtaining an alternate representation based on the DFT

For the wavelet algorithm, each functional block has a dual representation in the frequency domain. The dual in the frequency domain of the e_{00} block is simply an index-by-index multiplication of the DFT of the input block and the filter coefficients. Using properties of the finite-field discrete Fourier transform [7], it is possible to represent the structure containing the e_{00} block and the $z^{-\frac{M-1}{2}}$ cyclic shift from Figure 3.5 as shown in Figure 3.17. The vector $x^{(i)} = x_e^{(i)}|x_o^{(i)}$ is the input to the i^{th} elementary decryption block (likewise for the output vector y).

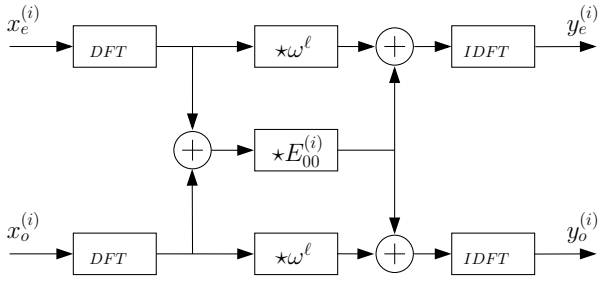


Figure 3.17: DFT alternative representation of the elementary decryption block

Call the block in Figure 3.17 the DFT wavelet decryption block. In Figure 3.17, the DFT and associated IDFT blocks represent the M-point discrete Fourier transform and its inverse, respectively. The block labelled $[\star E_{00}^{(i)}]$ represents the element by element multiplication with the DFT coefficients of the former e_{00} block. The block labelled $[\star\omega^\ell]$ is the dual of the $z^{-\frac{M-1}{2}}$ cyclic shift operation, which is the element by element multiplication with ω^ℓ, where $\ell = -\frac{M-1}{2}$. Define the vector $\omega^k = [\ \omega^0\ \omega^k\ \omega^{2k}\ \cdots\ \omega^{(M-1)k}\ ]$. The parameter ω is an element of order M in $\mathbb{GF}(256)$.

However, considering the upsampling and downsampling functions within the elementary decryption block, we represent two cascaded elementary decryption blocks with Figure

3.18. The block labelled $[\star\omega^1]$ is the due to the upsampling, downsampling, and associated delays. It is possible to factor out the $[\star\omega^1]$ blocks so that the cascade of elementary wavelet decryption blocks can be represented with the cascade of consecutive DFT decryption wavelet blocks and the appropriate number of multiples of the $[\star\omega^1]$ block, along with the necessary DFT and IDFT blocks.

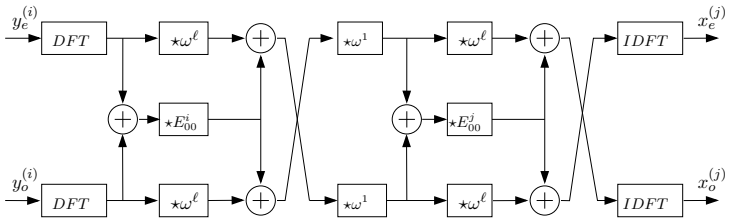


Figure 3.18: Cascade of two elementary decryption blocks of the DFT alternative representation.

This alternate representation changes the amount of diffusion that occurs between the DFT and IDFT blocks. Each of the functions between the DFT and IDFT are index by index operations, thus each symbol of y is only dependent upon the symbol of x of the corresponding index. This pushes all of the diffusive dynamics of the block to the outermost functions, DFT and IDFT. Since the DFT and IDFT cancel each other out in the cascade of two consecutive DFT decryption wavelet blocks, no diffusion occurs from one block to another. The next section describes the method to attack the wavelet system with this alternative representation.

3.4.5.2 Analysis of the DFT-based Attack

It is possible to attack the two-round system with the DFT representation. By using the branch originating from $C^{(m)}$ in Figure 3.16, we will try to recover the key. We consider the alternative DFT representation of the $C^{(m)}$ expression ($\mathbf{C}$ =$[C^{(m)}\ 0\ 0]$), which is 4 cascaded DFT wavelet decryption blocks followed by 4 multiples of the $\star\omega^1$ function inside the DFT and IDFT blocks. This attack recovers the individual key symbols by way of an exhaustive search on one set of filter coefficients, $e_{00}^{(1)}$ and using a derived expression to obtain $e_{00}^{(2)}$.

The chosen ciphertext $\mathbf{C}$ is $[C^{(m)}\ 0\ 0]$, where the odd and even indices of $C^{(m)}$ are such that $DFT(C) = C_e|C_o$. $C^{(m-1)} = 0$ and $C^{(m-2)} = 0$. With some manipulation we can

show that the plaintext $P = P_e|P_o$ can be described by

$$P_e = IDFT(w^{4\ell+4} \star C_o + w^{3\ell+4} \star (C_o + C_e) \star (E_{00}^{(4)} + E_{00}^{(3)} + E_{00}^{(2)} + E_{00}^{(1)})) \quad (3.21)$$

$$P_o = IDFT(w^{4\ell+4} \star C_e + w^{3\ell+4} \star (C_o + C_e) \star (E_{00}^{(4)} + E_{00}^{(3)} + E_{00}^{(2)} + E_{00}^{(1)})) \quad (3.22)$$

Also note that the sum of the $E_{00}^{(i)}$'s can be written in terms of known values, defining this expression, $\Sigma_E = [\sigma_{E(0)} \cdots \sigma_{E(\frac{M-1}{2})}]$ to be

$$\Sigma_E = E_{00}^{(4)} + E_{00}^{(3)} + E_{00}^{(2)} + E_{00}^{(1)} = \frac{DFT(P_e) - w^{4\ell+4} \star C_o}{w^{3\ell+4} \star (C_o + C_e)} \quad (3.23)$$

Note that where $E_{00}^{(1)}$ and $E_{00}^{(2)}$ are the key dependent vectors that are chosen independently, the vectors $E_{00}^{(3)}$ and $E_{00}^{(4)}$ are dependent upon $E_{00}^{(1)}$ and $E_{00}^{(2)}$, respectively.

Each index of Σ_E can be expressed by

$$\sigma_{E(i)} = E_{00}^{(1)}(i) + E_{00}^{(2)}(i) + E_{00}^{(3)}(i) + E_{00}^{(4)}(i) \quad (3.24)$$

By observing (3.24), we can verify that, by a judicious choice of the permutation (that we described in Section 3.3.2.3) the attacker cannot do better that an exhaustive search on all of the key values. Hence, we conclude that the two-round wavelet cryptosystem is secure against the DFT-based attack.

3.4.6 Summary of the attacks on WBC

Of the attacks on the wavelet algorithm presented thus far, Table 3.1 lists the computation complexity required for the attacks. The attacks which use the method of solving nonlinear systems of equations can be improved. These results show that the two-round wavelet cryptosystem is secure against all of the proposed attacks.

Table 3.1: Computational complexity of cryptanalytic attacks on WBC

Attack	**Complexity**
Exhaustive Key Search	$\mathcal{O}(2^{128})$
Divide And Conquer Interpolation	$\mathcal{O}(2^{158})$
Delta Function	$\mathcal{O}(2^{307})$
DFT-based	$\mathcal{O}(2^{128})$

3.5 Computational complexity analysis of WBC

This section provides a comparison of the wavelet cryptosystem with two well known cryptographic algorithms, DES and AES. The measures in Table 3.5 have been taken from the NESSIE report [81] for DES and AES. We also include the results for the two-round wavelet cryptosystem.

For our implementation, we have chosen to use the Montgomery arithmetic algorithms [56] for the $\mathbb{GF}(256)$ multiplications and inverse operation required in the wavelet algorithm. However, for our results in Table 3.5, we have provided a comparison of the algorithms where the multiplication is considered to be a table look-up. In this way, all three algorithms consist of table look-ups and logical operations. We claim that the operations required in each algorithm give an indication of the potential speed in software and hardware. Obviously, this metric is not the ultimate comparison as optimizations are made when implementing each algorithm. However, WBC shows promise of being highly optimal in DPSs in that there is a long history of implementing filter banks in DSPs, as mentioned previously. As we expected, the complexity of the wavelet decryption is half that of the wavelet encryption. The number of multiplications and logical operations for the wavelet algorithm as being superior to those of DES and comparable to AES [81]. The last two rows show that WBC possesses lower numbers of Table Lookups ('TLUs per bit') and comparable Logical Operations ('Logical OPs per bit') when normalized per bit in each algorithm. WBC does not use complicated structures or operations. Hence, the system can be implemented by filters (shift registers) and a simple inverse operation, which results in its efficient computational complexity.

3.6 Summary

In this chapter, we provided an overview of the first application of finite-fields wavelets to cryptography. A methodology, which we attribute to being work proposed by Fekri, is presented that constructs cryptographic primitives by using nonlinear wavelets. One interesting aspect of the proposed system is that it is easy to analyze. We studied both the efficiency and security of the developed system. We show that the wavelet encryption

Table 3.2: Complexity of operations for three block ciphers

Algorithm	**DES**	**AES**	**WBC Enc**	**WBC Dec**
Rounds	16	12	2	2
Block size (bits)	64	128	240	240
Word size (bits)	32	8	8	8
Key size (bits)	56	128	128	128
Table lookups/ Table size (bits x bits)	123/8(6×4), 8/8(8×32), 64/11(8×48), 16/16(8×64), 0/8(8×56)	160(8×32)	184/(256×1)	122/(256×1)
Shifts/Rotations + multiplications	0	30	124	124
XOR, ADD (bit size)	6 (32 bit) 64 (48 bit), 14 (64 bit)	11 (128 bit), 120 (32 bit)	1732 (8 bit)	926 (8 bit)
Total TLUs (8 bit)	216	160	60	60
Total logical OPs (8-bit)	520	656	1732	926
TLUs per bit	3.375	1.25	0.767	0.508
Logical OPs per bit	8.125	5.125	7.216	3.86

system with effective key size of 128 possesses computational complexity lower than DES and comparable to AES with key sizes of 64 and 128, respectively. It is also imperative to note that the wavelet decryption has almost half the complexity of the encryption. However, the complexity of the decryption of DES is the same as the encryption and that of AES is 30% slower than the encryption. Therefore, the complexity of the wavelet decryption is lower than both DES and AES decryption. We investigated classical and new cryptanalytic attacks against the wavelet cryptosystem. WBC does not possess strong linear or differential characteristics. Furthermore, we developed variants of interpolation, delta function, and discrete Fourier transform-based attacks, where we showed that these attacks cannot do better than the exhaustive key search method. Wavelet-based cryptography presents a new approach to cryptographic primitives that demonstrates promise for use in wireless hand-held devices.

PART II

Network Security

CHAPTER IV

SECURE CONNECTIVITY IN WIRELESS SENSOR NETWORKS

4.1 Introduction

Secure communications and connectivity in wireless sensor networks are crucial requirements of network properties for many proposed applications of these networks [3]. The requirements of networks are such that reliability and efficiency are of paramount importance. It is also necessary that the performance of these networks not sharply degrade as a result of any security measures implemented in the network. Furthermore, it is worthwhile to analyze the potential threat presented by an adversary to the network connectivity. It is also important for the network to optimize the limited resources available in each of the individual nodes, which also optimizes the global performance of the network. In this chapter, we study the performance of networks with regard to global network connectivity and secure communications.

We first look at the required communication range for nodes in a network to provide connectivity. It is vital to provide each node a means to communicate with all nodes in the network. While certain networking situations only require communication with the base station, we envision a net-centric environment where the network may be highly dynamic and reconfigurable to adjust to the flow of communications. The connectivity property is a measurement of such an ability. We consider the connectivity of networks while employing key management schemes that provide secure communications to the network. We also observe increased demands on the resources of the nodes with the inclusion of security measures to these networks. Here, we consider the key distribution scheme as the security service. Additionally, we consider an adversarial presence in the network, and we examine the effect that malicious attacks have on the network.

As the adversary gains control of nodes in the network, it is able to degrade connectivity in the network by partitioning the network or isolating nodes as a result of capturing

nodes. With the accumulation of key information, it may be possible for the adversary to compromise links elsewhere in the network. This is a result of the necessity of the wireless sensor network to employ a key predistribution scheme as opposed to the desired private key infrastructure (PKI), which offers a greater measure of security [28, 85]. However, the limitations of computational ability and direct communication with a trusted third party render these PKI systems infeasible for wireless sensor networks. In PKI, each link created is independent of other links in the network. In key predistribution schemes, there is an overlap of key information among the established links. Thus, the adversary may compromise a particular link without directly compromising either of the two nodes involved. The adversary is able to eavesdrop and decrypt all transmissions for each compromised link. The problem with the physical vulnerability of the sensor nodes is that the key information for the key predistribution scheme is also vulnerable. In this work, we provide the first analysis of global connectivity with respect to network parameters by considering key management schemes and adversarial attacks. Current work focuses on studying connectivity with regard to communication range and link-compromise in the context of key management schemes.

In this chapter, we study connectivity and the rate of link compromise by considering these network properties while considering the global state of the network. We combine the effects of communication range, key predistribution schemes and node-compromise attacks into a single expression to evaluate connectivity in wireless sensor networks.

4.1.1 Network model

We employ the following notation throughout this chapter. We consider a network of n nodes randomly distributed into a field of unit area. Each node has communication radius r or $r(n)$, where a uniform radius is assumed. For node-compromise attacks, we consider either the number of compromised nodes, x, or the fraction of the network compromised, p_c. For the sake of analysis, we define the following probabilities in Table 4.1. When considering p_k, this is the probability of establishing a secure key based on the initial network parameters upon deployment. The rate of link-compromise, p_ℓ, is a function of p_c. In this work, we do not consider node failures as a result of non-malicious actions. The purpose of studying p_{sf}

Table 4.1: List of network probabilities to define wireless sensor network

p_e	Probability of establishing a link.
p_{sf}	Probability that a sensor functions properly.
p_k	Probability two nodes can establish a secure link.
p_c	Fraction of the network compromised by the adversary.
p_ℓ	Probability of link-compromise.
p_{conn}	Probability of network connectivity.

is in the context of adversarial attacks on the network; therefore in this work, $p_{sf} = 1 - p_c$.

We use random graphs and their properties to represent wireless sensor networks. A graph is connected if there is a connected path of nodes and edges between any two nodes. For clarity, we define several random graphs to represent wireless sensor networks in different scenarios. First, we consider the simple geometric graph, $G(n, r)$, which models a wireless sensor network of n nodes with a link or edge between two nodes if the distance between them is within communication range, r. For network schemes that employ a key distribution scheme, we consider the secure link graph, $G(n, r, p_k)$. In this model, there is an edge between two nodes that are within communication range, r, with probability p_k, as defined by the key predistribution scheme.

We also consider graphs representing networks that are deployed in adversarial environments, where an adversary is attacking the network by randomly capturing a fraction p_c of the nodes in the network. We define the compromised secure connectivity graph to be $G(n, r, p_k, p_c)$. The case where the compromise of nodes is not random is not considered in this chapter.

We note some relationships between these defined network models. There are differences between the connectivity and the secure link graph when nodes begin to become compromised. In the geometric graph $G(n, r)$, any node that is removed from the network affects only links that are incident with the compromised node. However, for the compromised secure link graph $G(n, r, p_k, p_c)$, the removal of the fraction p_c of the nodes in the network additionally removes p_ℓ edges from the network. These links are indirectly compromised

as defined by the security model that we employ. The security model we follow is that each node possesses a finite number of cryptographic keys that are used to create secure links between nodes. The capture of a node results in the compromise of all of the key information in the captured node. Accumulation of compromised nodes and their associated key information results in the compromise of links elsewhere in the network, where the key information used to establish a particular link has been acquired by the adversary. Therefore, in $G(n, r, p_k, p_c)$, the rate of link-compromise p_ℓ is a function of the fraction of nodes compromised. Determining this relationship with regard to connectivity is one of the contributions in this chapter.

4.1.2 Overview of contribution

We explore the relationship between network properties with respect to the resources expended by the individual nodes and the effect of malicious attacks on the network. The contribution of this work is a metric that enables a fair comparison of particular instances of wireless networks employing key management schemes where there is an adversary present in the network. The metric compares the relative resource usage of each node required to maintain a certain network property. In this work, we consider global connectivity and consider the measure of required communication radius to compare various instances of wireless sensor networks. While sensor nodes do not necessarily support dynamic transmission power, we use this as a measure of the resource usage of the network to compare with other instances or conditions of the network. While this chapter does not propose a new key predistribution scheme, this work proposes a metric that is used to compare the resilience of network resources employing different key predistribution schemes and to examine how they react in the presence of adversarial attacks.

Section 4.2 establishes a baseline approach to consider the simple case of global network connectivity and identify its relationship to communication range. We also consider both simple connectivity graphs $G(n, r)$ and secure connectivity networks $G(n, r, p_k)$. This establishes a practical network scenario in terms of the resource usage required to implement key predistribution schemes on a wireless network. These expressions can be used

to compare the resource cost of $G(n, r, p_k)$ with $G(n, r)$ or the difference in resource cost between various instances of key predistribution schemes. Additionally, we consider the same problem in response to an adversarial node compromise attack. In Section 4.3, we study the relationship between p_c and p_s, p_e on the $G(n, r, p_k, p_c)$. We derive an expression that considers key distribution schemes and node-compromise attacks, which is used to determine the communication range required for network connectivity. Last, Section 4.4 introduces the resiliency-connectivity (RC) metric, a metric that measures the resilience of wireless networks employing key management schemes in the presence of adversarial attacks by observing the communication range required for global connectivity. The resilience of the required communication radius reflects the overall resources consumed by the network in these particular scenarios to maintain the global property of connectivity. The notable contribution of this work is the establishment of the important relationship between connectivity and security in wireless sensor networks. One aspect of this topic that we do not consider is the problem of adversarial node detection and revocation. This is a difficult problem, but we consider this to be a separate problem of study. We allow ourselves to assume that the network can determine which nodes have been compromised from the network. Additionally, compromised nodes cannot masquerade as legitimate nodes in the network.

4.1.3 Related work

We review several previous results in areas related to the connectivity property in wireless networks. This chapter involves the study of networking properties including connectivity, key management schemes, and adversarial models in sensor networks. The composite relationship between these networking properties is explored in this chapter. We consider different key predistribution schemes and extend recent developments regarding the connectivity property for wireless sensor networks.

In terms of connectivity, Gupta and Kumar [43, 44, 99] present the classical work on wireless networks. This work generates results for connectivity on the graph $G(n, p)$. Pishronik [79] derives expressions for connectivity in large-scale sensor networks. Rather than

using $G(n,p)$, this work considers $G(n,r,p_e,p_s)$, since the $G(n,r,p_e,p_s)$ graph is argued to be a more appropriate representation of wireless sensor networks and their associated limitations. An expression is derived to specify the minimum communication radius in order for the network to be connected by

$$r(n) \geq \sqrt{\frac{\ln n}{n p_{sf} p_e \pi f_{min}}} \quad \text{as} \quad n \to \infty. \tag{4.1}$$

The expression includes the minimum node density of the distribution of the nodes in the network f_{min}. We consider nodes in a network that are distributed into a field in a uniformly random deployment; therefore, $f_{min} = 1$ in this work. A network of these parameters will be connected with high probability if the communication radius is greater than $r(n)$ as defined by (4.1). This work also states that connectivity not only occurs at this value of $r(n)$, but also exhibits a sharp threshold effect.

There are several proposed methods for key management schemes for wireless sensor networks. Because of their large-scale deployments, a promising approach for establishing secure links within networks in a randomly deployed network is the idea of key predistribution. Details of these works are described in Section 2.3.3. Eschenauer and Gligor [35] provide one of the original works on key predistribution schemes for sensor networks. Chan [13] follows up on this scheme by proposing a q-composite scheme where q common keys are required to establish a secure link. We note that the Eschenauer scheme is equivalent to the q-composite scheme when $q = 1$. Du et al. [32] proposed a scheme that combines the random key predistribution approach with the classical method of Blom's key predistribution [8]. The goal of this approach is to increase the resilience of the network to node-compromise attacks without increasing memory usage. Also, Liu and Ning [60] develop a pairwise key establishment scheme using a polynomial-based key predistribution protocol and probabilistic key distribution methods. This scheme suggests advantages over other schemes by having keys with a threshold property. Additionally, the rate of compromised links as a result of node compromises in the network also behaves with this threshold effect. Delgosha [26] proposes a key predistribution scheme based on multivariate symmetric polynomials. Also, there are key predistribution methods that take advantage of deployment knowledge.

Du et al. [33] and Liu and Ning [60, 61] propose schemes that exploit deployment knowledge to increase network performance in terms of connectivity and memory usage.

We consider the two schemes of multivariate symmetric polynomial key predistribution [26] and random key predistribution [35] in our work to investigate the resilience of networks to node-compromise attacks. Additionally, we demonstrate the relevance of this work for networks of varying size, so we consider large- and small-scale networks. For the q-composite scheme, a networking instance is defined by the value of q, the size of the key ring, and the size of the key pool. We denote this scheme by $\text{QCOMP}(n, q, m, P)$. For the multivariate polynomial scheme, one instance of this scheme is defined by the dimension of the key space b and the degree of the polynomials used, t, so we have $\text{MKPS}(n, b, t)$. In the interest of space and clarity, this work consists of simulation and analytical results pertaining to only the q-composite and multivariate symmetric polynomial key predistribution schemes.

4.2 Communications range and connectivity

We first consider $G(n, r)$ and determine the communication range that every node needs to transmit in order to have global network connectivity. As stated earlier, this establishes the baseline networking case. According to (4.1), the $r(n)$ required for connectivity in $G(n, r)$ is

$$r(n) \geq \sqrt{\frac{\ln n}{n\pi}}. \tag{4.2}$$

This is shown in Figure 4.1, where the probability of connectivity is plotted versus the communication range $r(n)$ of the nodes in the network. The plot for $G(n, r)$ for $n = 5000$ is the solid line plot.

We also consider the networks employing key predistribution schemes as modelled by the secure connectivity graph, $G(n, r, p_k)$. The parameters for each key predistribution scheme determine p_k. The term p_k in (4.3) is the p_e in (4.1) since the probability of establishing a link is only dependent on the key predistribution scheme in this networking situation. We will see later that p_e also considers the fraction of compromised links, p_ℓ. Therefore, we can represent the required communication range for this network model, given n and p_k, by

$$r(n) \geq \sqrt{\frac{\ln n}{np_k\pi}}. \tag{4.3}$$

Parameters between the two key predistribution schemes were chosen such that the memory usage in each individual sensor node is identical. We define one key in the QCOMP to be equivalent in memory size to one polynomial coefficient in the MKPS. In [26], it states that the memory usage is $b(t+1)\mathbb{F}$, where $\mathbb{F}$ is the size of the field being used. Therefore, by assuming each key in the key ring for the q-composite scheme key ring is from the field $\mathbb{F}$, we set $m = b(t+1)$.

The required communication range for connectivity plots for the geometric graph and secure connectivity networks are shown in Figure 4.1. The probability of connectivity of the network versus the required communication radius is plotted for QCOMP$(5000, 1, 20, 1000)$ and MKPS$(5000, 2, 9)$. Based on the parameters of each key distribution scheme, $p_{k(\text{MKPS})} = 0.0278$ and $p_{k(\text{QCOMP})} = 0.3350$. According to (4.3), the threshold value $r(n)$ for the q-composite scheme is $r(n) = 0.0402$ and for the multivariate scheme is $r(n) = 0.1397$. Figure 4.1 shows the results of p_{conn} versus $r(n)$ for $G(n, r)$, QCOMP$(5000, 1, 20, 1000)$, and MKPS$(5000, 2, 9)$ determined by simulation. It seems that the QCOMP network appears to be more resource efficient than the network, as the required communication radius for QCOMP is almost one-third of the MKPS communication radius. However, we see that this is not necessarily the case, as is investigated in the following sections.

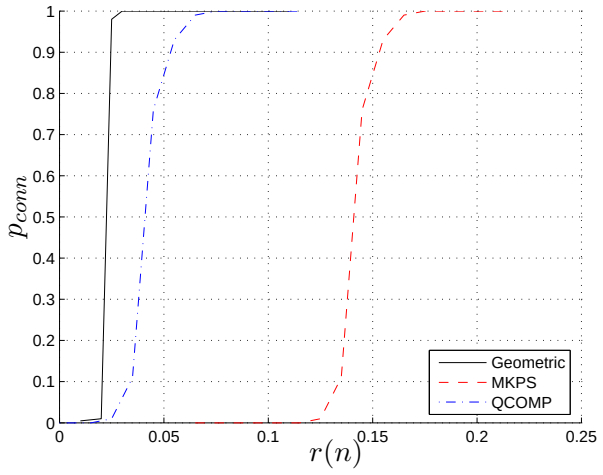


Figure 4.1: p_{conn} vs. $r(n)$ for $G(n, r)$, QCOMP$(5000, 1, 20, 1000)$ and MKPS$(5000, 2, 9)$.

4.3 Connectivity with node-compromise

After establishing the baseline connectivity requirements in networks employing key predistribution schemes, we now consider the presence of an adversary in the network. A fraction p_c of the nodes in the network is compromised by the adversary. This section examines the effect of the node-compromise attack and its influence on connectivity among other network properties. We identify which key predistribution schemes provide greater resilience to node compromise attacks.

For networks employing key management schemes, we consider the network model $G(n, r, p_k, p_c)$. In this model, for each compromised node, its key information is also compromised. The adversary can use its set of compromised keys to compromise links elsewhere in the network. A link in the network is compromised if the adversary has obtained all the keys used to create the secure link. It is necessary to investigate the relationship between p_c and the probability of link-compromise, p_ℓ. Where p_ℓ is a function of p_c, it is also dependent on the key predistribution scheme.

To adapt (4.3) to this networking situation involving key predistribution schemes and node-compromise attacks, we include p_c and p_ℓ in our analysis to define the expression

$$r(n) \geq \sqrt{\frac{\ln n}{n(1-p_c)p_k(1-p_\ell)\pi}}. \tag{4.4}$$

This section describes how each of these parameters affects the network. We study the communication range requirements to provide connectivity to a network as a function of the fraction of compromised nodes p_c. It is possible to provide analysis with (4.4). In this section, we analyze the relationships between the fraction p_c of the network compromised and both p_ℓ and p_{sf}. As stated previously, the probability of a sensor properly functioning is defined by $p_{sf} = (1 - p_c)$. Additionally, we represent p_e with both p_ℓ and p_k, as these two network parameters affect the probability that a link is formed. At the same time, p_ℓ and p_k are independent of each other, so they can be represented as the product $p_e = (1 - p_\ell)p_k$. While we can determine p_k from the initial network parameters, the relationship between p_c and p_ℓ with regard to $G(n, r, p_k, p_c)$ depends on the key predistribution scheme.

We now describe the relationship of p_c to p_ℓ by the choice of the key predistribution

scheme. We need to analyze key predistribution schemes and determine the rate at which links are compromised in the network. First, this involves detailing the rate at which key information is compromised as a function of the number of nodes compromised in the network. Then, the rate of link compromise can be determined by the rate at which key information is obtained by the adversary. These expressions can be inserted into (4.4). After these rates are identified, it is possible to create the analysis between node-compromise attacks and the rate of link-compromise to network connectivity in Section 4.4.

4.3.1 Rate of link-compromise for $MKPS$

For MKPS, each node possesses b key shares for each dimension of the key space. Since the identities of two nodes that have established a secure link differ in one index, the link between the two nodes can be compromised if the $b-1$ common polynomials are recovered by the adversary.

We note that the order t of the polynomials used in the network determines the ability of the adversary to compromise a polynomial in this scheme. In order to recover the coefficients of the polynomial, the adversary needs $\binom{t+b-1}{b-1}$ key shares of a polynomial to have enough information. Therefore, increasing t simply increases the resilience of this scheme against adversarial attacks while only increasing the memory usage for each node. However, one can see that varying t does not affect p_k. As stated in Section 6.2.1, the authors determine the probability of link-compromise in $p_{\ell(\text{MKPS})}$ in (2.11), which is the compromise of $b-1$ polynomials for a specific link.

4.3.2 Rate of link-compromise for $QCOMP$

In the case of the q-composite key predistribution scheme, the rate of link-compromise is a function of the fraction of the key pool that is compromised. It can be shown that the number of keys compromised from the network P' is a function of the fraction p_c of compromised nodes and network parameters m, P by the expression

$$P' = P\left(1 - \left(1 - \frac{m}{P}\right)^{p_c n}\right). \tag{4.5}$$

After determining the number of compromised keys as a function of the number of compromised nodes, we determine the rate of compromised links as a function of the number of compromised nodes. In this scheme, the adversary is able to compromise a secure link if all the q keys in the secure link have been obtained. The expected number of compromised links is a function of the number of compromised keys. The rate of link-compromise is determined by the network parameters q, P, P' in the expression

$$p_{\ell(\text{QCOMP})} = \frac{\binom{P'}{q}}{\binom{P}{q}}. \tag{4.6}$$

We note that the total number of secure links present in the network is $\lceil n(\pi r^2 n)p_k \rceil$, where $\lceil \pi r^2 n \rceil$ is the average degree of a node in the network. With respect to security, it is desirable for p_ℓ to increase as slowly as possible. Along with p_c, these are the two adversarial parameters that have a direct influence on overall network connectivity.

Additionally, we note one caveat is that this study allows for the q-composite key predistribution scheme. In this key predistribution scheme, two nodes may share more than q keys. The original q-composite scheme creates a session key based on q of the set of shared keys. Compromise of the q keys used to create that particular secure link results in the compromise of the link. This study allows for nodes to recognize which nodes and subsequently which keys to remove from their own key ring. Two nodes can establish a new link with the remaining shared keys. For example, two nodes in a q-composite network (with $q = 2$) share four keys $\{k_1, k_2, k_3, k_4\}$. If the initial session key is $k = f(k_1, k_2)$, and if k_1 and k_2 are compromised, then the link is compromised according to the original q-composite scheme. For this work, we allow a new key k' to be established as a new session key, where $k' = f(k_3, k_4)$.

4.4 Resiliency-Connectivity metric analysis

While the probability of link-compromise directly determines the increased adversarial control of the network, this also affects the connectivity properties of the remaining legitimate nodes. Obviously, one strategy in the securing of these networks is to exclusively concentrate on a network that minimizes the probability of link failure resulting from the compromise

of key information elsewhere in the network.

One extreme approach is to have each node carry a unique key for every other node in the network. The compromise of any number of nodes does not reveal any information about any other links in the network. However, prohibitive amounts of memory usage in each node make this scheme impractical, as each node in the network must carry $n-1$ keys. On the other extreme, a network could implement a key management scheme using a global key to minimize storage requirements for each node, where the same key is used in every link in the network. However, any node compromise results in the compromise of all key information in the network, which is not desirable.

Therefore, when considering resource-constrained devices, there is a compromise between employing key predistribution schemes, which result in a high probability of secure link establishment, and those that establish resilient links in the face of node compromise attacks. Existing work contains comparisons between other key predistribution schemes to attempt to justify improvements on previous schemes by separately considering the rates of link-compromise and secure link establishment. We provide a composite analysis of these two vital network parameters that have thus far been analyzed separately.

The relationships that have been previously analyzed are the fraction of the network compromised and the probability of link compromise (p_c vs. p_ℓ) and the relationship of the probability of secure link establishment and overall network connectivity (p_k vs. p_{conn}). Here, we want to combine these two relationships to ultimately find the relationship between node compromises and overall network connectivity (p_c vs. p_{conn}). We call the result of this analysis the resiliency-connectivity (RC) metric, which is illustrated in Figure 4.2. The RC metric analysis determines the resilience of a network in terms of connectivity with an adversarial presence as a function of required communication radius $r(n)$. This resilience is a representation of the relative amount of resources required by the nodes in the network in order to maintain global connectivity.

With our analysis of $G(n, r, p_k, p_c)$, it is possible to examine network connectivity as a function of the fraction of compromised nodes in the network. With regard to connectivity, we consider the global connectivity within a deployment of a sensor network. This is the

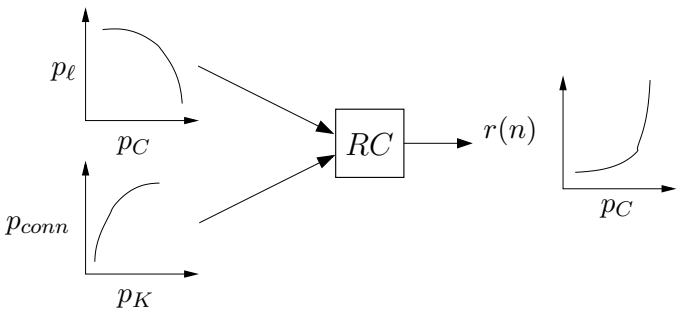


Figure 4.2: Illustration of the resiliency-connectivity (RC) metric.

first study that considers node compromises in wireless sensor networks and the influence on global secure network connectivity. We establish the RC metric to compare various instances of wireless sensor networks using key predistribution schemes. This is a measure of the ability of the network to maintain an adequate number of secure links to have global secure connectivity given an increasing p_c. From a first look of the RC metric, one could simply look at the connectivity property of the network as a function of the fraction of the network that is compromised. However, we choose to consider the required communication radius for our analysis, as noted in Figure 4.2.

With the assumption that sensor nodes can dynamically adjust their transmission range, the rate at which nodes within a network have to expend additional resources to maintain connectivity can determine an ordering between networks in terms of their resilience to node-compromise attacks. For example, the two baseline key management schemes (global key and pairwise keys) have the two extreme resilience performance metrics. For a network using a global key, $p_\ell = 1$ for any value of $p_c > 0$. Therefore, the communication range required for connectivity is infinite once a single node is compromised. This follows from the idea that all links are compromised with the compromise of one, since any node holds the key information for the entire network. On the other hand, a network employing the pairwise key scheme is optimally resilient in that for each node that is compromised, the adversary gains no additional information about links elsewhere in the network. Thus, for the pairwise scheme, $p_\ell = 0$. However, for other key predistribution schemes, we can analyze the the RC metric by using our analysis of secure connectivity with node-compromise attacks. From

(4.1) we derive (4.4) to examine the RC properties of the secure connectivity graph by

$$r(n) \geq \sqrt{\frac{\ln n}{\pi(1-p_\ell)(1-p_c)np_k}}. \tag{4.7}$$

It is noted that n and p_k are network design parameters that are independent of p_c, while p_ℓ is a function of p_c. So, it is possible to characterize the required communication radius $r(n)$ with (4.7) as a function of the network design parameters and p_c.

We can investigate the increasing required communication radius as the number of compromised nodes increases. A desirable networking scenario is such that $r(n)$ increases minimally as p_c increases. An increasing $r(n)$ can be interpreted by the idea that as the network is losing nodes and links, each node is forced to attempt to communicate with more nodes in order to establish global connectivity. The p_ℓ of a networking scheme more resilient to node-compromise attacks increases at a slower rate than others, which results in $r(n)$ increasing at a slower rate. We also note that there are instances where the network will never be connected regardless of what range the nodes can transmit (resulting in $r(n) \to \infty$). In the compromised secure connectivity graph $G(n, r, p_k, p_c)$, the new communication range value $r'(n)$ from (4.4) can be written in terms of the initial $r(n)$ specified from $G_S(n, r, p_k)$ and (4.3). With high probability, the compromised secure connectivity graph $G(n, r, p_k, p_c)$ is connected again with communication radius $r'(n)$ specified by

$$\frac{r'(n)}{r(n)} \geq \frac{\sqrt{\frac{\ln n}{np_k(1-p_c)(1-p_\ell)\pi}}}{\sqrt{\frac{\ln n}{np_k\pi}}} = \sqrt{\frac{1}{(1-p_c)(1-p_\ell)}}. \tag{4.8}$$

We note that the ratio of the increased communication range in the networks with node compromises is illustrated in the following two sections, 4.4.1-2. The RC metric is represented by showing the relationship between $r(n)$ and p_c. Section 4.4.1 contains Figures 4.4, 4.5, and 4.6, which are the RC metric results for large-scale networks. Section 4.4.2 has Figures 4.10 and 4.11 to show the simulation results for the RC metric for small-scale networks. For small-scale networks, we show that the asymptotic analysis described by (4.4) fails to accurately determine the required communication range for small-scale networks. Despite the inaccuracies of the asymptotic model for connectivity of small-scale networks, we observe that the comparison of the two key predistribution schemes demonstrates behavior

similar to the large-scale networks.

It is worth mentioning that this approach may not be efficient or optimal in certain situations or applications. There are instances where every node may not need to communicate with the transmission power as stipulated by (4.4) and still have a global connectivity. Equivalently, there may be cases where, if a small subset of nodes increases its communication range by some value, connectivity is reestablished. Furthermore, dynamically adjusting the communication range may not be possible for nodes in certain networks, or the adjustment may be too costly in terms of energy usage.

4.4.1 RC metric for large-scale networks

This section considers the RC metric with regard to large-scale wireless networks using analytical expressions derived thus far. Using (4.7), we have found the required communication radius as a function of the number of compromised nodes for networks of 5000 nodes. In these cases, each network has been created using one of the two key predistribution schemes. For example, the RC metric concept is illustrated in Figure 4.3 where $r(n)$ versus p_c is plotted for a large-scale network simulated with the q-composite predistribution scheme, QCOMP$(5000, 1, 30, 5000)$. The increase of $r(n)$ is shown as a function of an increasing fraction of the network becoming compromised, p_c. With this analysis, it is possible to compare instances of secure networks and determine the relative resilience of the networks to node-compromise attacks. In the case of the RC metric, the more resilient network with respect to connectivity is the one that requires less transmission radius over values of p_c. The interpretation of the RC metric is that the required communication range is a measure of the resources required to maintain global connectivity in the network. The RC metric addresses the expected increase in resources required to maintain connectivity in the presence of node compromise attacks. This section illustrates the use of the metric by comparing the resilience of QCOMP and MKPS with various comparable parameters.

For consistent and proper comparison between networks, we consider wireless networks of sensor nodes where each node stores the same amount of key information in memory. Additionally, we consider the case where the initial probability of secure link establishment, p_k

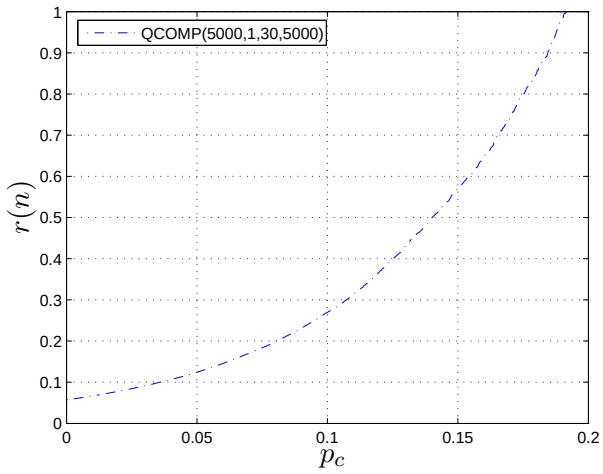


Figure 4.3: $r(n)$ required for connectivity vs. p_c for QCOMP$(5000, 1, 30, 5000)$.

for each network is the same. Furthermore, this results in an initial required communication range $r(n)$ being the same for each key predistribution scheme. The parameters for the key predistribution schemes are chosen accordingly. For example, we consider three networks in MKPS$(5000, 2, 9)$, QCOMP$(5000, 1, 20, 14200)$, and QCOMP$(5000, 2, 20, 1500)$. Based on the derivations of $p_{k(\text{QCOMP})}$ and $p_{k(\text{MKPS})}$ and the parameters of each network, $p_k = 0.0278$ for each of the networks. Additionally, each of the nodes in these networking instances is set to $m = b(t + 1) = 20$. Figure 4.4 is a plot comparing the RC metric of three network instances in MKPS$(5000, 2, 9)$, QCOMP$(5000, 1, 20, 14200)$, and QCOMP$(5000, 2, 20, 1500)$. First, the QCOMP network where $q = 2$ possesses an inferior RC metric compared to the other two networks. From the plot, the MKPS network performs better with regard to the RC metric for values of approximately $p_c < 0.15$. However, the QCOMP network where $q = 1$ is more resilient for $p_c > 0.15$. Between QCOMP networks of $\{q = 1\}$ and $\{q = 2\}$, the $\{q = 2\}$ network has a key pool that is considerably smaller than the key pool for the $\{q = 1\}$ network. With the same memory, the link-compromise rate occurs at a much faster rate since a greater fraction of the key pool is compromised with each increase in p_c. With regard to the MKPS network, its behavior in demonstrating greater resilience than the QCOMP network for low values of p_c can be attributed to the threshold property in its link security.

We also show a comparison of the RC metric for instances of large-scale networks for

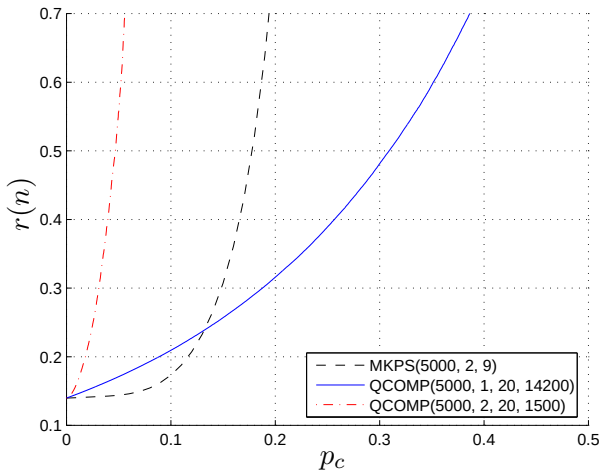


Figure 4.4: $r(n)$ required for connectivity vs. p_c for QCOMP$(5000, 1, 20, 14200)$ and QCOMP$(5000, 2, 20, 1500)$ and MKPS$(5000, 2, 9)$.

both the MKPS and QCOMP schemes, MKPS$(5000, 3, 9)$, QCOMP$(5000, 1, 30, 103000)$, and QCOMP$(5000, 2, 30, 6325)$. Figure 4.5 shows three networks that possess equal initial values of p_k and require the same amount of memory in each node. This illustration demonstrates the effect on the RC metric by modifying parameters of the respective key predistribution schemes. As compared to Figure 4.4, the RC metric for MKPS$(5000, 3, 9)$ compared to MKPS$(5000, 2, 9)$ shows significant improvement of the RC metric as a function of p_c compared to its QCOMP network counterpart. The RC metric of the MKPS network performs better than that of the QCOMP network for values of $p_c < 0.5$. The increase in the dimension of the key space for the MKPS network provides an increase in link security of the key predistribution scheme. Increasing the b parameter pushes the threshold of $p_{\ell(\text{MKPS})}$, where links become compromised at higher values of p_c. Furthermore, we show the RC metric for MKPS with $b = 4$ and its corresponding QCOMP network in Figure 4.6. This network instance takes advantage of the t-secure property, as described in [26]. Regardless of the magnitude of p_c, $p_{\ell(\text{MKPS})} = 0$ since there is not a sufficient number of key shares in the network to recover the polynomial coefficients. The MKPS$(5000, 4, 9)$ network outperforms the QCOMP$(5000, 1, 4, 330000)$ network with regard to the RC metric.

These examples demonstrate the utility of the RC metric. It is possible to use the RC

metric to compare the performance of any key predistribution schemes in terms of node-compromise attacks.

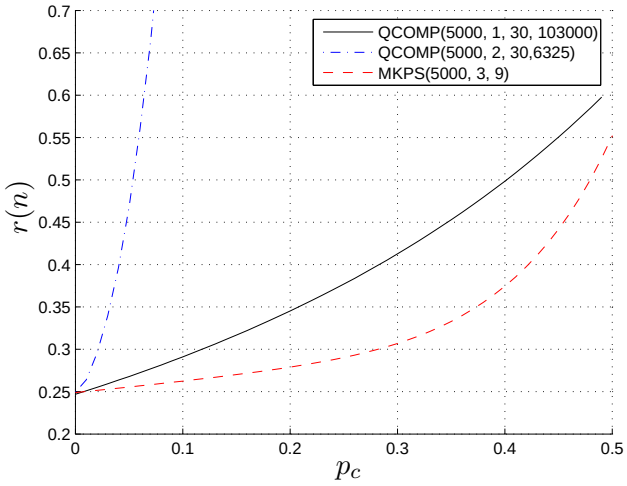


Figure 4.5: RC metric results for MKPS(5000, 3, 9), QCOMP(5000, 1, 30, 103000), QCOMP(5000, 2, 30, 6325).

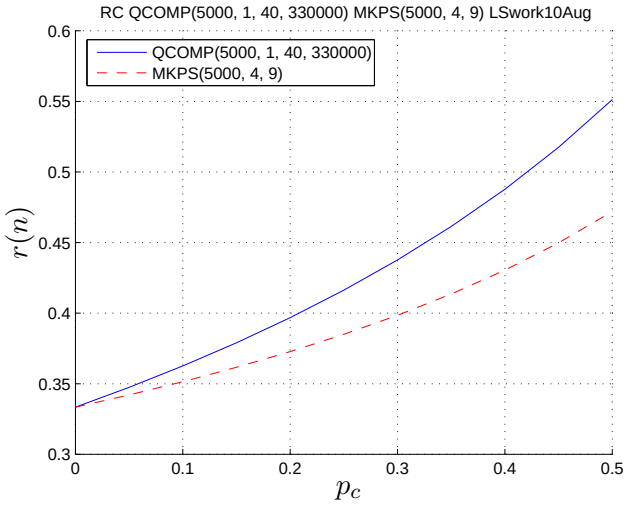


Figure 4.6: RC metric results for MKPS(5000, 4, 9), QCOMP(5000, 1, 40, 330000).

4.4.2 RC metric for small-scale networks

This section investigates the viability of the RC metric for small-scale networks. The analytical results that we have shown with respect to the RC metric are valid for large-scale

sensor networks, as these are asymptotic results (for $n \to \infty$). However, for small n, the analytical results in [79] are not valid. Figure 4.7 shows the plot of the probability of connectivity versus communication radius $r(n)$. The two small-scale networks that are plotted are MKPS$(500, 2, 9)$ and QCOMP$(500, 1, 20, 4620)$. Note that the value of p_k and memory usage are equivalent for both networks. Although the network begins to become connected with non-zero probability at the asymptotic threshold for connectivity, it is evident that for small-scale networks, the threshold effect is not present or an accurate measure of connectivity. This behavior that makes the asymptotic results invalid for small scale networks can be attributed to boundary effects and a sparse network deployment. Therefore, we cannot depend on these results to provide accurate estimates of the connectivity property for these networks. In this section, we aim to perform security and connectivity analysis similar to the large-scale networks and show their relevance to small-scale networks. Our justification for the connectivity behavior will be in the form of simulations.

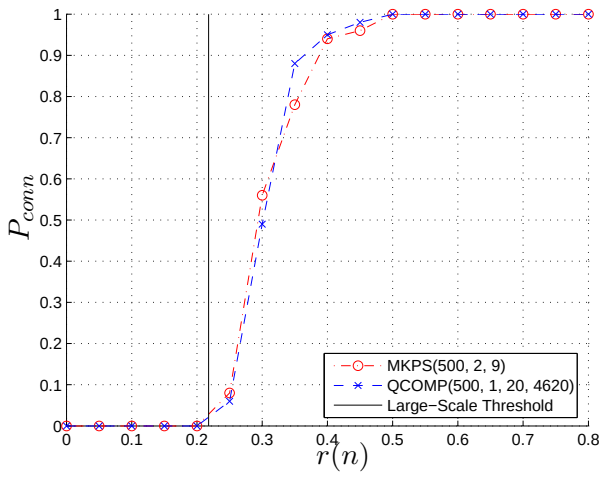


Figure 4.7: p_{conn} vs. $r(n)$ for MKPS$(500, 2, 9)$ and QCOMP$(500, 1, 20, 4620)$ and the theoretical asymptotic connectivity threshold.

For small-scale networks, we have implemented actual instances of the key predistribution schemes in simulation. In analyzing the RC metric for small-scale networks, we determine the communication radius $r(n)$ that is required to establish connectivity among

the remaining legitimate nodes in the network. The simulations were performed by removing p_c of the nodes as well as their associated compromised links. Then, the communication radius was increased incrementally until global connectivity was established in the network. We show the probability of connectivity versus p_c for QCOMP$(500, 1, 20, 4620)$ and MKPS$(500, 2, 9)$ in Figure 4.8. This analysis was was performed along with the simulations to determine the required communication radius in compromised secure networks.

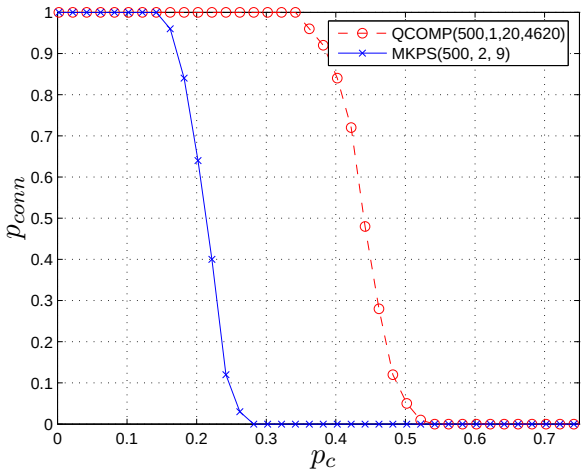


Figure 4.8: p_{conn} vs. p_c for QCOMP$(500, 1, 20, 4620)$ and MKPS$(500, 2, 9)$.

Furthermore, we look at the RC metric simulation results for small-scale networks. First, Figure 4.9 illustrates the disparity of the actual RC metric performance of a network defined by QCOMP$(500, 1, 20, 4620)$ versus the theoretical result. It is evident that the large-scale analysis does not accurately portray the actual behavior of the RC metric for small-scale networks.

We show the justification of the viability of the RC metric for small-scale networks through the simulation of two instances of QCOMP and MKPS networks. Figure 4.10 shows the comparison of networks of 500 nodes and the plots of $r(n)$ as a function of p_c for two instances of compromised secure connectivity networks. The solid line with x's is the plot for MKPS$(500, 2, 9)$ and the dashed line with circles represents the plot for QCOMP$(500, 1, 20, 4620)$. We also note that there is a dashed-dotted line extrapolating the MKPS network simulation. As with the large-scale networks, the memory usage and initial p_k values in each node for

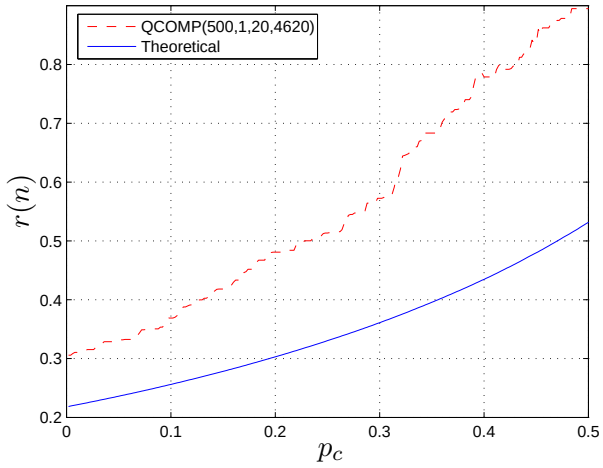


Figure 4.9: Comparison of actual RC metric performance and theoretical asymptotic result for QCOMP$(500, 1, 20, 4620)$.

the key predistribution schemes have been set to be the same. In this case, $m = 20$ and the initial value of $p_k = 0.0832$. We can observe that the MKPS network is more resilient than the QCOMP network for node-compromise values up to $p_c = 0.25$, where the network is not connected with high probability. This comparative resilience between the two key predistribution schemes is similar to that of the large-scale model results. The MKPS scheme demonstrates a more desirable performance of the RC for low values of p_c, but soon gives way to the QCOMP scheme for $p_c \approx 0.25$.

The second instance of the RC metric for small-scale networks is in Figure 4.11, which shows the RC metric for two networks in MKPS$(500, 3, 9)$ and QCOMP$(500, 1, 30, 21550)$, where $p_k = 0.04$ and $m = 30$. The behavior of the two key predistribution schemes for small-scale networks matches the relative performance of the RC metric for large-scale networks. When increasing the dimension of the key space (the parameter b) in the MKPS network, this yields a considerable increase in the RC metric. The RC metric result of MKPS for $b = 3$ demonstrates a greater resiliency and RC metric performance than for the associated QCOMP network. We note that the sacrifice of the networks in Figure 4.10 and those in Figure 4.11 is the initial required communication radius $r(n)$.

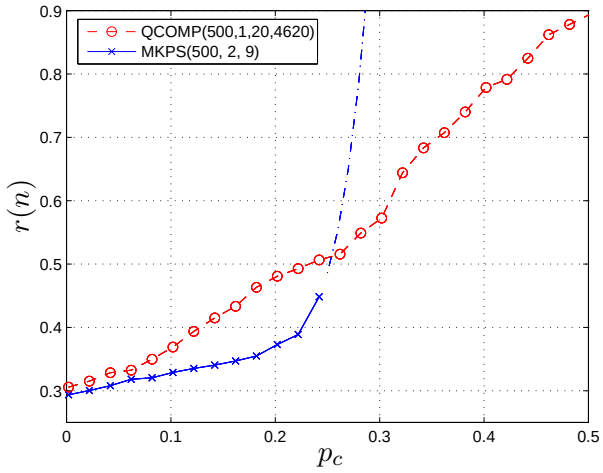


Figure 4.10: $r(n)$ required for connectivity vs. p_c for MKPS$(500, 2, 9)$ with extrapolation and QCOMP$(500, 1, 20, 4620)$.

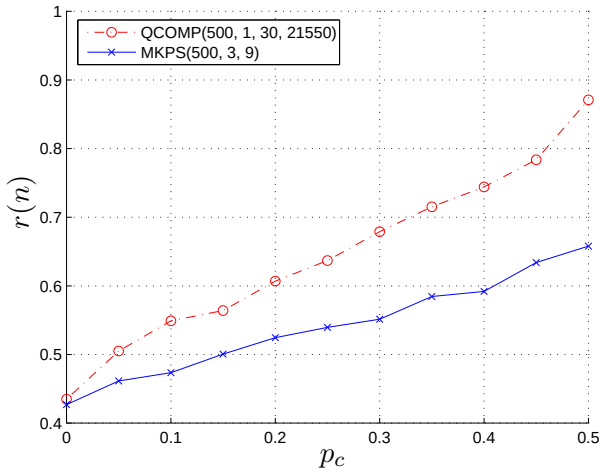


Figure 4.11: $r(n)$ required for connectivity vs. p_c for QCOMP$(500, 1, 30, 21550)$ and MKPS$(500, 3, 9)$.

4.5 Summary

This work has analyzed the effect of node-compromise attacks on sensor networks and its interaction with key predistribution schemes and secure network connectivity. Investigating the communication range required for connectivity is the approach we used to compare the resilience of networks to node-compromise attacks. We have established a relationship between the communication range and key predistribution parameters for large-scale networks through analysis and simulation. Then, we established the relationship between node-compromise attacks and the communication range required for connectivity in large-scale networks. With these relationships, we introduced the resiliency-connectivity (RC) metric, which is a measure that determines the resilience of the connectivity for a network in the presence of an adversarial entity. Additionally, we have extended these results to small-scale networks through simulations. The resilience of the network measures the ability of a particular network instance to maximize properties relating to connectivity while minimizing the usage of limited resources throughout the network.

CHAPTER V

RELATING NETWORK LATENCY AND THE RESILIENCE OF KEY PREDISTRIBUTION TO NODE-COMPROMISE ATTACKS

5.1 Introduction

We analyzed the resilience of the global connectivity of wireless sensor networks using key predistribution in the presence of an adversary in the previous chapter. In this chapter, we consider network properties involving the transmission of information amongst the nodes. The analysis of these network properties includes the temporal aspect of networking scenarios, whereas our analysis of network connectivity only considered the resilience of static network properties. The temporal aspect of the network scenario allows for the analysis of packets being transmitted throughout the network. This results in a dynamic situation as the set of nodes that are actively transmitting changes with time. In order to consider the dynamic nature of the network, it is necessary to incorporate network services to deal with the flow of packets. For this work, we define a medium access control (MAC) protocol, a routing protocol, and a traffic model. As in Chapter 4, where we considered the effect of node compromise attacks on secure network connectivity, we consider the effect that key predistribution and adversarial attacks have on the ability of the nodes in the network to deliver packets to a specific destination (a sink node). We analyze the average packet latency for various networking parameters and also establish a measure of achievable throughput for these networks. The goal of this work is to consider packet latency and network throughput in wireless sensor networks employing key predistribution schemes. Further, we study these issues when the network is under adversarial node-compromise attack.

We expect that the behavior of the delivery of packets to the sink to perform similarly to the results for connectivity in Chapter 4. In the previous chapter, we determined the communication radius of the nodes required for the network to be connected. As derived in Section 4.2 with (4.3), networks with equivalent n, r and p_k, have the same requirement for

communication range to provide connectivity. In the same way, we expect that networks with similar parameters to provide similar networking performance in terms of average packet latency and achievable throughput. For example, if we consider the networking scenario where events occur in one location of the network and need to be sent to another area of the network, the packets containing this information will be transmitted in a multi-hop fashion through the network until it reaches its destination. Depending on several parameters which include routing and channel access protocols, the information will arrive at its destination after some delay in time, or latency. Given networks with equivalent n, r, and p_k, these networks will, on average, have a comparable number of links to transmit information through the network. Thus, we expect networks with the same p_k to perform similarly in terms of the average time it takes to transmit packets through the network. We explore this property of packet latency and achievable throughput with respect to different key predistribution schemes for wireless sensor networks.

Furthermore, we consider latency and throughput in networks that are being attacked by an adversary. As shown in Chapter 4, the node compromise attacks not only remove nodes from the network, but links are also compromised as the adversary accumulates key information from the compromised nodes. As the links and nodes in the network are compromised, it is expected that the average packet latency increases and the packet reliability degrades. We examine this behavior on wireless sensor networks that are using key predistribution schemes.

Existing work with regard to packet latencies in wireless sensor networks has considered the idea of energy efficiency via sleep scheduling [69, 31, 101]. The lifetime of the network is increased by having each node periodically enter a *sleep* state, which reduces the energy consumption from processing and transmitting across the network. There are strategies to optimize the transitions between the *sleep* and *active* states. In the same way, we consider latency given a fraction of the network compromised by an adversary. For a network in an adversarial environment, one can consider some fraction of the nodes and links to be removed from the network. Instead of nodes being unavailable while being in the *sleep* state, nodes and links are unavailable as a result of being compromised.

We compare the resilience of the maximum achievable throughput or average packet latency as a function of the magnitude of the adversarial attack for different key predistribution schemes. The resilience of wireless sensor networks and their key predistribution schemes to node compromise attacks can be measured by the increase in packet latency or the decrease in maximum achievable throughput. We present results by providing simulations for networks comparing the QCOMP and MKPS key predistribution schemes. First, we examine the average packet latency and packet reliability as a function of adversarial influence in the network. Second, we determine the maximum achievable throughput of the networks with respect to adversarial attacks. Last, we examine average packet latency as a function of the distance from the sink node. Considering packet latency at different distances from the sink illustrates the resilience of the network, given the key predistribution scheme, from another perspective. We also consider packet latency in network instances of MKPS and the variation of the dimension parameter, d. We determine that the MKPS key predistribution provides a more resilient network against the node-compromise attack compared to networks using QCOMP.

5.2 Network model

We consider a network of n nodes uniformly randomly deployed into a field of unit area. Each node has communication radius r and establishes a secure link with a neighbor with probability p_k, as defined by the key predistribution scheme used. We consider and compare the behavior of QCOMP and MKPS. In this networking situation, there is a sink node placed in the center $(0.5, 0.5)$ of the unit area. We investigate the network while it is operating in a data-gathering scenario. The nodes in the network detect events and generate corresponding packets to be sent to the sink node in a multi-hop fashion.

In terms of network performance, the latency of a packet is the time between the time that the packet is generated to the time that it reaches its destination. Also, packet reliability is the probability that a packet being sent will actually reach its final destination. We define average packet latency, $\mathbb{t}$ to be the average packet latency of all packets generated from all nodes of the network. Likewise, average packet reliability, designated as ρ, is the

fraction of all packets reaching their final destination successfully. Measurements from an MKPS network are indicated by a subscript MKPS, $\mathbb{t}_{\text{MKPS}}$, ρ_{MKPS}. For matters of analysis in this chapter, we define $\mathbb{t}_{\Delta}$ to represent the difference between the average packet latency values for the QCOMP and MKPS key predistribution schemes ($\mathbb{t}_{\Delta} = \mathbb{t}_{\text{QCOMP}} - \mathbb{t}_{\text{MKPS}}$). We define several protocols necessary for the study of latency and throughput in networks given an insecure deployment environment.

- **MAC protocol**: The MAC protocol that we use in this chapter is a simple two-hop ready-to-send/clear-to-send (RTS/CTS) protocol based on [82]. The protocol has three steps: RTS phase, CTS phase and a transmit (TX) phase . The time it takes to complete one round of these three phases is defined to be one *time epoch*. Packet latencies in this work will be measured in time epochs. In the RTS phase, each node who has a packet in its buffer to send, transmits an RTS packet. Each RTS packet is sent at a random time, and the length of the RTS window is chosen to be sufficient to have RTS collisions with low probability. In the CTS phase, each node responds with a CTS to at most one requesting neighbor node. Nodes that receive a CTS from all of its neighbors are permitted to send a packet in the TX phase. The protocol allows all transmitting nodes to send without any of its two-hop neighbors interfering with its transmission.

- **Packet transmission model**: As stated previously, we are measuring the ability of networks to transmit packets in a data-gathering scenario. A node in the network detects a particular event and generates information packets to report it back to the sink node in a multi-hop fashion. We define an event to be represented by one packet randomly generated at a random node. The generation of these events is determined by a Poisson process with a network-wide arrival rate of λ packets per time unit. Each node maintains a packet buffer for storage of unsent packets.

  This rate of event generation, λ, can also be considered as the throughput for a particular network instance. We also define the maximum achievable throughput to be the λ for which a network is able to provide a $\mathbb{t}$ and average packet buffer length that

is not growing unboundedly. For the maximum achievable throughput of a network, we define this property to be λ_{max}. We define the difference in maximum achievable throughput between QCOMP and MKPS by $\lambda_\Delta = \lambda_{\text{QCOMP}} - \lambda_{\text{MKPS}}$.

In actual network operations, we expect that the generation of events are correlated. Nodes in close proximity presumably detect the same event. In addition to the geographic correlation to event generation, these events will be temporally correlated. However, this is left for future consideration, as we solely consider the even generation to occur uniformly randomly throughout the network. Additionally, events are only generated at nodes that are still part of the network, so there are no events being generated at compromised nodes.

- **Routing protocol**: Given the data-gathering task of the network, we consider routing protocols to relay the generated packets to the sink node. A straight-forward method to accomplish this is to use a geographic routing approach [53, 57, 72, 84, 88, 90, 103]. Packets are routed through the network by forwarding the packet progressively closer to the sink node based on position information of the nodes. We also assume that nodes will be able to find a legitimate node to forward packets, if one exists.

- **Adversarial model**: The adversary we consider is the node compromise attacker, the same as in Chapter 4. The attacker randomly compromises nodes in the network, accumulating nodes and the keys found in these nodes. Any link in the network is compromised if the adversary has obtained every key used to establish that particular link. Nodes and links that are compromised are no longer available for the network to use. The fraction of the network compromised is p_c and the rate of link compromise is p_ℓ. The resulting random graph model we consider is the compromised secure connectivity graph, $G(n, r, p_k, p_c)$.

These various protocols and models have been proposed to be suitable for use in wireless sensor networks. Although there may be more attractive alternatives, the goal of this research is to analyze key predistribution schemes applied to wireless sensor networks and

their resilience to node-compromise attacks. By using the same protocols for each key predistribution alternative, we can compare the relative performance between key predistribution schemes. There are no biases towards any of the key predistribution schemes for any of these defined network protocols.

5.3 Packet transmission in networks with node-compromise attacks

We have simulated several networking scenarios to present the resilience of packet latency and achievable throughput in networks with regard to node-compromise attacks. We consider a network of $n = 500$ nodes and fix a communication radius that is sufficient to provide initial connectivity in the network according to the results in Section 4.4.2. The networks use either QCOMP and MKPS, where both the memory used in each node and the probability of secure link connectivity between two nodes is equivalent. We examine node-compromise attacks on the network that will compromise secure links formed by the respective key predistribution schemes. In Table 5.1, we consider several pairs of networks using QCOMP and MKPS with parameters set to have equal p_k.

Table 5.1: Network parameters for QCOMP and MKPS key predistribution for latency simulations.

QCOMP$(\mathbf{n}, \mathbf{q}, \mathbf{m}, \mathbf{P})$	p_k
QCOMP$(500, 1, 20, 4620)$	0.083
QCOMP$(500, 1, 30, 21550)$	0.041
QCOMP$(500, 1, 40, 61880)$	0.025

MKPS$(\mathbf{n}, \mathbf{d}, \mathbf{t})$	p_k
MKPS$(500, 2, 9)$	0.083
MKPS$(500, 3, 9)$	0.041
MKPS$(500, 4, 9)$	0.025

5.3.1 Resilience of packet latency to node compromise attacks

We see that the resilience of packet latency behaves similarly to the rate of link compromise in these networks. The first results we present are the average packet latencies,$\mathbb{t}$, for the networks as a function of the fraction of the network compromised, p_c. Figure 5.1 and Figure 5.2 show the average packet latency and packet reliability as function of p_c for

MKPS$(500, 2, 9)$ and QCOMP$(500, 1, 20, 4620)$, where $p_k = .083$. The event traffic rate is $\lambda = 0.10$. In this instance, the communication range of the nodes is chosen to be $r(n) = 0.25$. As nodes and links from the initial routes become compromised, packet latency increases and packet reliability begin to degrade. Packet latencies are increased as nodes are forced to route through suboptimal paths. Packet reliability begins to degrade as some of the nodes are unable to find a path to route their packets back to the sink node.

We note that in Figure 5.1, the initial average packet latencies are equal $\mathbb{t}_{\text{QCOMP}} = \mathbb{t}_{\text{MKPS}}$. This matches our results from Chapter 4 for secure connectivity with $G(n, r, p_k)$. This result verifies the property that two wireless sensor networks with comparable n, r, and p_k provide equal average packet latencies in the data-gathering scenario. The result in Chapter 4 that this matches is the communication range required for network connectivity being the same for networks of equal n and p_k. The property of having equal initial $\mathbb{t}$ is also present in Figures 5.3 and 5.5.

Given an initial $\mathbb{t}$, we are now able to consider the resilience of the network to node-compromise attacks. Specifically, we are interested in looking at the rate of link-compromise in the key predistribution schemes for these networks and its effect on packet latencies. As seen in Figure 5.1, the average latency of MKPS grows unboundedly at $p_k = 0.25$ while QCOMP is able to provide bounded latency until $p_k = 0.35$. This matches the resiliency-connectivity results, where $r(n) \to \infty$ for MKPS$(500, 2, 9)$ at $pc \approx 0.15$. This relationship to the the packet latency is that as $r(n)$ approaches infinity, the network loses its connectivity. This ultimately results in the latency to increase.

Plots are shown in Figures 5.3 and 5.4 for the network instances of MKPS$(500, 3, 9)$ and QCOMP$(500, 1, 30, 21550)$, where $p_k = .083$ and $\lambda = 0.10$. Here, $r(n)$ is chosen to be 0.45. Simulation results are also shown for MKPS$(500, 4, 9)$ and QCOMP$(500, 1, 40, 61880)$, where $p_k = .025$. The arrival rate of the events is $\lambda = 0.10$ in Figure 5.5 and Figure 5.6. For the third network scenario, $r(n)$ is chosen to be 0.65. These results match the resiliency-connectivity results in Chapter 4, as evident from the results in Figures 5.1 - 5.6. The resilience of the rate of link-compromise against the node-compromise attack is consistent with a smaller increase in average packet latency as a function of p_c.

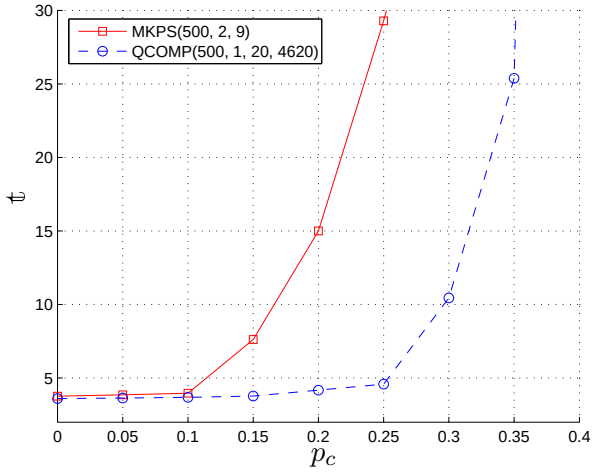


Figure 5.1: $\mathbb{t}$ vs. p_c for MKPS$(500, 2, 9)$ and QCOMP$(500, 1, 20, 4620)$ with $\lambda = 0.10$ and $r = 0.5$.

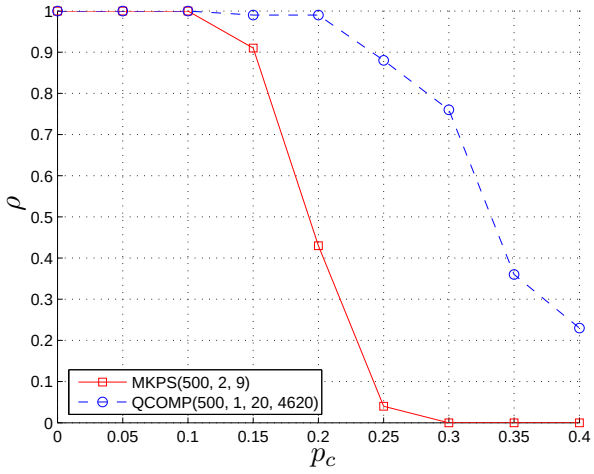


Figure 5.2: ρ vs. p_c for MKPS$(500, 2, 9)$ and QCOMP$(500, 1, 20, 4620)$ with $\lambda = 0.10$ and $r = 0.5$.

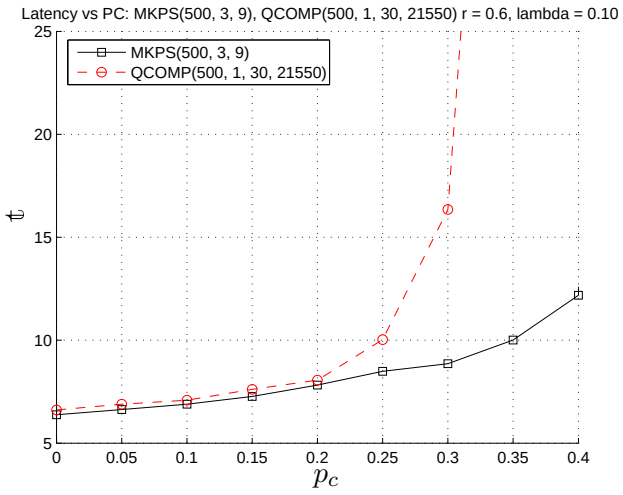


Figure 5.3: $\mathbb{t}$ vs. p_c for MKPS$(500, 3, 9)$ and QCOMP$(500, 1, 30, 21550)$ with $\lambda = 0.10$ and $r = 0.6$.

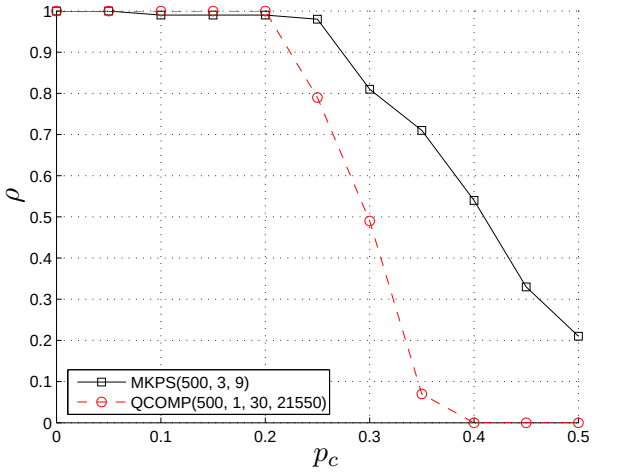


Figure 5.4: ρ vs. p_c for MKPS$(500, 3, 9)$ and QCOMP$(500, 1, 30, 21550)$ with $\lambda = 0.10$ and $r = 0.6$.

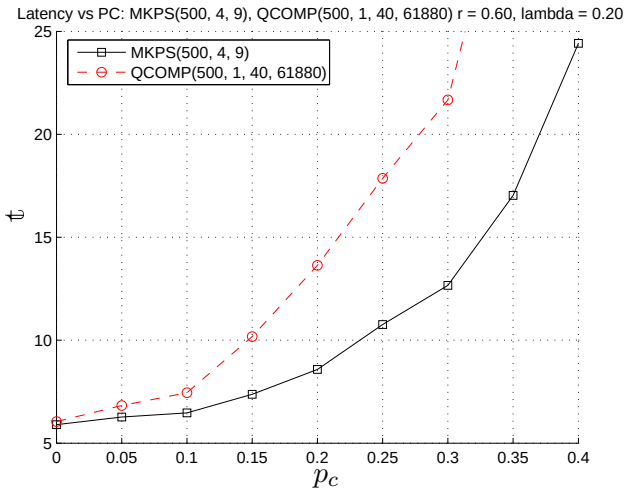


Figure 5.5: $\mathbb{t}$ vs. p_c for MKPS$(500, 4, 9)$ and QCOMP$(500, 1, 40, 61880)$ with $\lambda = 0.20$ and $r = 0.75$.

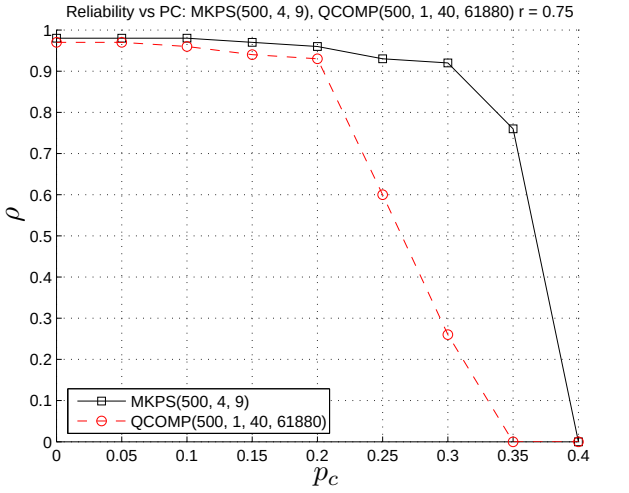


Figure 5.6: ρ vs. p_c for MKPS$(500, 4, 9)$ and QCOMP$(500, 1, 40, 61880)$ with $\lambda = 0.20$ and $r = 0.75$.

5.3.2 Resilience of maximum achievable throughput to node compromise attacks

Using the results from Section 5.3.1, we measure the maximum throughput, λ_{max}, that a network is able to provide bounded packet latency. Given $\lambda > \lambda_{max}$, the node buffers begin to increase as events are generated faster than the network can route packets to the sink. We determine λ_{max} for the three instances of QCOMP and MKPS.

Figures 5.7, 5.8, and 5.9 show λ_{max} versus p_c for the three pairs of networks that we have considered. These plots show the increased resilience of λ_{max} to node-compromise attacks for the MKPS scheme for when $d = \{3, 4\}$ as compared with QCOMP. In Figure 5.7, the QCOMP network is able to handle the same λ_{max} for values of p_c that are 0.10 higher than the MKPS network. However, in Figures 5.8 and 5.9, this property shows that MKPS where $d = \{3, 4\}$, the λ_{MAX} is more resilient to node-compromise attacks than QCOMP. Given increased link resilience, it is expected that the maximum achievable throughput in these networks demonstrates similar gains in performance when comparing MKPS with QCOMP.

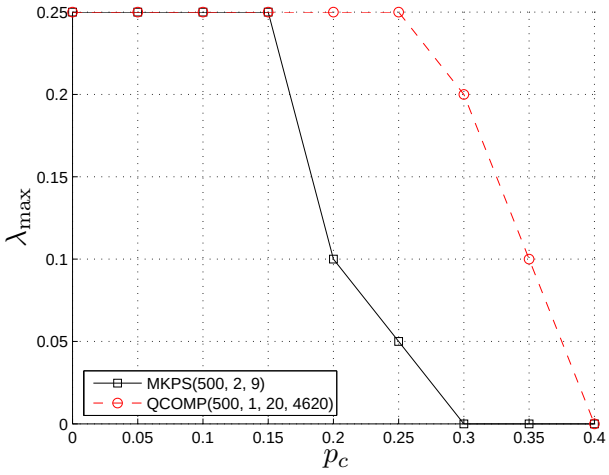


Figure 5.7: $\lambda_{\max}$ vs. p_c for MKPS$(500, 2, 9)$ and QCOMP$(500, 1, 20, 4620)$ with $r = 0.5$.

5.3.3 Relationship between latency resilience and node location

We present another interpretation of measuring packet latency as a function of node compromise attacks. In this section, we consider packet latency as a function of the distance

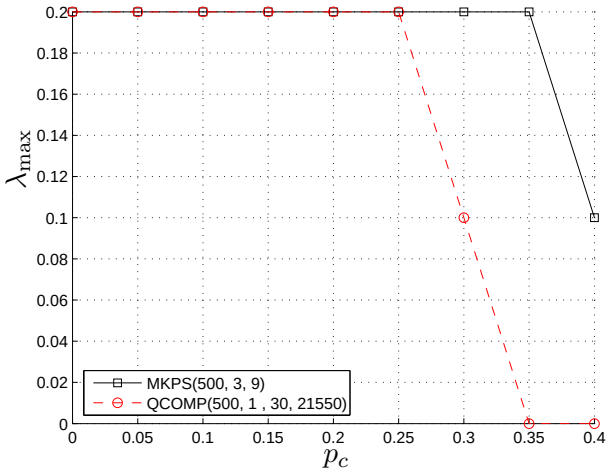


Figure 5.8: $\lambda_{\max}$ vs. p_c for MKPS$(500, 3, 9)$ and QCOMP$(500, 1, 30, 21550)$ with $r = 0.6$.

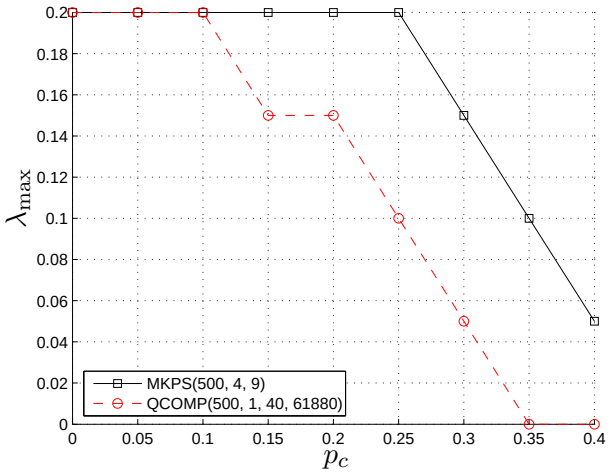


Figure 5.9: $\lambda_{\max}$ vs. p_c for MKPS$(500, 4, 9)$ and QCOMP$(500, 1, 40, 61880)$ with $r = 0.75$.

from the sink, d_{sink}. In normal networking situations, it is expected that the latency of packets originating farther away from the sink will be higher than those closer to the sink node. We consider the average packet latency of events generated at nodes in 0.1-unit increments from the sink node. Our results do, in fact, show a relative increase in packet latency as we consider regions in the network farther from the sink node. We include the node compromise attack into consideration of packet latency as a function of distance. This section analyzes the average packet latency of packets originating from a specific distance from the sink and obtaining relative measures of the resilience of the latency of these packets as a function of p_c.

We examine packet latency for specific instances of networking situations. In Figure 5.10, we consider QCOMP$(500, 1, 30, 21550)$ and MKPS$(500, 3, 9)$ with $\lambda = 0.05$ and $p_c = 0.10$. For all distances from the sink node, $\mathbb{t}_\Delta > 0$, indicating that MKPS is slightly improved compared to QCOMP at these parameters. However, in the case of MKPS where $d = 4$, there is a significant gain with respect to $\mathbb{t}_\Delta$, as shown in Figure 5.11. For MKPS$(500, 4, 9)$ and QCOMP$(500, 1, 40, 61880)$ with $\lambda = 0.05$ and $p_c = 0.20$, $\mathbb{t}_\Delta \geq 3$ for all distances from the sink node. The gain is dependent upon the values of p_c and λ. For instance, $\mathbb{t}_\Delta$ is greater in Figure 5.11 than Figure 5.10 because of the higher value of d, but also for the higher value of p_c. These results of packet latencies as a function of d_{sink} match our results pertaining to globally average packet latencies, $\mathbb{t}$.

5.3.4 Tradeoffs with MKPS key predistribution

We have seen results for the resilience of packet latency in wireless networks in response to node-compromise attacks. The MKPS key predistribution scheme was shown to provide improved resilience to node-compromise attacks. This section considers the tradeoffs when varying the dimension parameter, d. We have compared the performance of two MKPS networks with similar memory m, but varying p_k. Changing the dimension of MKPS effects p_k. Additionally, we have chosen $r(n)$ in each case to be sufficiently large enough to provide a connected network. We have chosen MKPS networks with the parameters listed in Table For this section, we use MKPS$(500, 2, 9)$ with $r = 0.5$ and MKPS$(500, 4, 4)$ with $r = 0.75$.

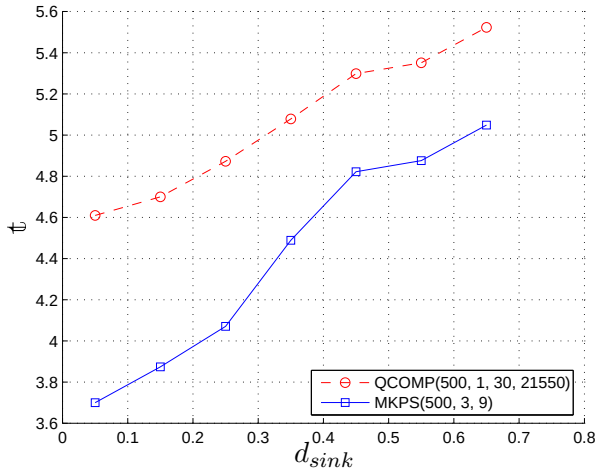


Figure 5.10: $\mathbb{t}$ vs. d_{sink} for MKPS$(500, 3, 9)$ and QCOMP$(500, 1, 30, 21550)$ with $\lambda = 0.05$ and $p_c = 0.10$.

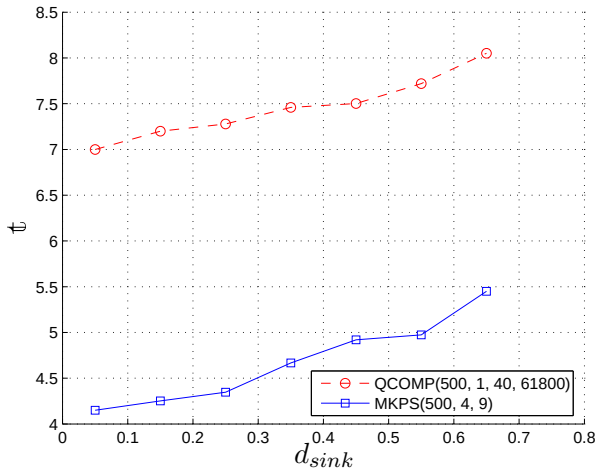


Figure 5.11: $\mathbb{t}$ vs. d_{sink} for MKPS$(500, 4, 9)$ and QCOMP$(500, 1, 40, 61800)$ with $\lambda = 0.05$ and $p_c = 0.20$.

The MKPS for $d = 2$ has $p_k = 0.083$, and for $d = 2$, it has $p_k = 0.025$. We note that $m = 20$ for both of these instances.

Increasing d results in a greater resilience to the node-compromise attack. Links are much more resilient to the node-compromise attack at an increased d; therefore, the network is able to stay connected and successfully route packets to the sink. This is shown in Figure 5.12, where $\lambda_{\max}$ vs. p_c is shown. However, the tradeoff is that the overall packet latency $\mathbb{t}$ is increased for higher dimension. In some cases $\mathbb{t}_\Delta = 2$. Figure 5.13 shows the difference in the average packet latency for the two networks for $\lambda = 0.05$. The dimension parameter allows a network to be designed for the maximum achievable throughput, $\lambda_{\max}$ to have an increased resilience against node-compromise attacks at the cost of an increased $\mathbb{t}$.

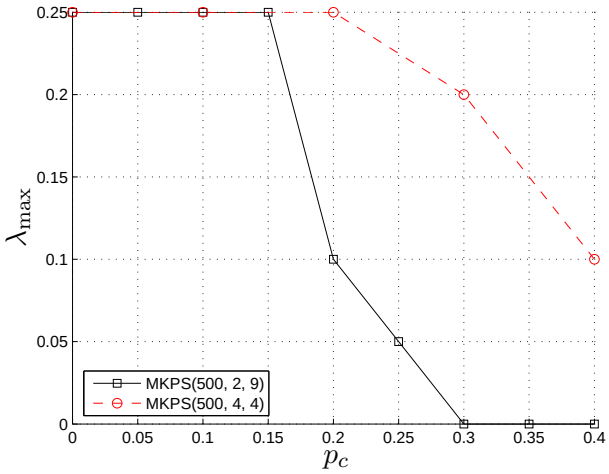


Figure 5.12: $\lambda_{\max}$ vs. p_c to compare the effect of the d parameter in MKPS.

5.4 Summary

In this chapter we have analyzed the resilience of packet latency in a data-gathering scenario. We extended the work of Chapter 4 to address the temporal aspects of networking to enable the measurement of average packet latency and maximum achievable throughput. We considered key predistribution schemes and wireless sensor networks and examined the resilience of latency and throughput as a function of node compromise attacks.

Furthermore, we compared MKPS and QCOMP and determined that MKPS for $d = \{3, 4\}$

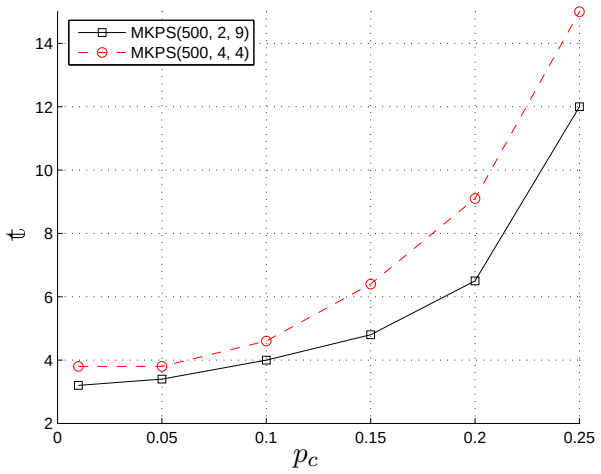


Figure 5.13: $\mathbb{t}$ vs λ to compare the effect of the d parameter in MKPS.

possesses superior resilience with respect to average packet latency and maximum achievable throughput. Also, we have analyzed packet latencies with respect to their distance from the sink node. We obtained similar results when comparing MKPS and QCOMP in this approach as well. Furthermore, the d parameter in MKPS increases the resilience of a network to node-compromise attacks with the cost of an increased average packet latency. This investigation provided an initial treatment of the resilience of packet latency and achievable throughput for key predistribution schemes applied to wireless sensor networks.

CHAPTER VI

KEY PREDISTRIBUTION SCHEMES AND THEIR RESILIENCE TO NODE-SPOOFING ATTACKS

6.1 Introduction

Thus far, we have considered issues pertaining to link security and secure connectivity in wireless sensor networks. These investigations have studied key predistribution techniques to provide secure keys for use in communication links in the network. Based on the key predistribution technique used, the networks offer some amount of resistance to node-compromise attacks in terms of link security. As the number of compromised nodes grows, the adversary increases its control over the links in the network. As the adversary compromises communication links in the network, he is able to eavesdrop on the transmitted data without having to directly compromise either of the end nodes of the link. Depending on the resources and capability of the adversary, the network is vulnerable to a class of more powerful attacks, ultimately allowing the adversary to have a greater effect on the reliability and availability of data in the network. A stronger malicious attack on sensor networks is if the adversary is able to masquerade as a legitimate node and perform malicious tasks without being detected. These tasks range from simple packet dropping and modification to the creation of worm holes and sink holes that alter the flow of packets through the network. This class of attack is called the node-spoofing attack. This chapter provides a study of node-spoofing attacks on the key predistribution schemes for wireless sensor networks.

We present an analysis of node-spoofing attacks in wireless networks for various key predistribution schemes. By considering various adversarial models for node-spoofing, we present new approaches to key predistribution by attempting to uniformly distribute the keys in the nodes of the network, which results in increased resilience against node-spoofing attacks. We present background details in Section 6.2, which include the network models

and notation used and a summary of work related to key predistribution schemes and security in wireless sensor networks. Section 6.3 presents several adversarial models considered in this chapter. These are specific to key predistribution schemes. We propose the regular key predistribution scheme and the regular threshold key predistribution scheme in Section 6.4. Analysis and simulations results are included in this section, which compares various proposed and existing key predistribution schemes.

6.2 Preliminaries

This section describes the network and key management models used in our analysis. Some of the notations used are described in Section 6.1. We assume a uniformly random deployment of n nodes into a field of unit area. Each node is capable of transmitting with communication radius r. In terms of key predistribution schemes applied to the network, each node is able to store m units of key information.

For the key predistribution scheme, there is a publicly-known pseudo-random function $f_{ID}(v_i) = K_i$ which maps the node identity to the key identities within each node. This is used after the neighborhood discovery phase, when the nodes establish session keys. Node identities are exchanged in the neighborhood discovery phase. A node v_i determines with which of its neighboring nodes, $v_j \in N(v_i)$, it can form a secure link by using $f_{ID}(v_j) = K_j \; \forall$ $v_j \in N(v_i)$ to determine if there are any shared keys. The identity space of $f_{ID}(\cdot)$ takes node identities of size n. For node-spoofing attacks in this chapter, we allow n node identities for the adversary to attempt to spoof. It is possible to design the network to recognize multiple $f_{ID}(\cdot)$ functions for future deployments.

With regard to the node-compromise attacks, we consider x compromised nodes or the fraction, p_c, of compromised nodes in the network. The security parameters we consider are the probability of link-compromise, p_ℓ, and the probability of node-spoofing, p_x. These probabilities are described in further detail in Section 6.3.

6.2.1 Related work

This section documents recent developments related to this work in this chapter. First, we state the proposed key predistribution schemes for wireless sensor networks that we consider

Table 6.1: Network parameters for node-spoofing attacks on wireless sensor networks

n	Number of nodes in the network
m	Number of memory units in each node
v_i	Node with identity i
$N(v_i)$	Neighborhood of node i
K_i	Key ring of node i
k_i^j	The jth key in the key ring of node i
$k_{ij} = k_{ji}$	Shared keys between node i and j
P	Key pool size
P'	Compromised key pool
$P_{N(v_i)}$	Key pool of neighborhood of v_i
λ	Key threshold
$\#(k)$	Number of occurrences of key k
$\text{BASE}(n, m, P)$	A network using random KPS
$\text{REG}(n, m, P)$	A network using regular KPS
$\text{TKEY}(n, m, \lambda, P)$	a network using threshold KPS
p_k	Probability of establishing a secure key between any two nodes
p_c	Fraction of the network compromised
p_ℓ	Probability of link-compromise
p_x	Probability of successful node-spoof

in this chapter. Second, we briefly cover the possible adversarial attacks once a node can be spoofed.

There are several proposed methods for key management schemes for wireless sensor networks that are used in this chapter. Eschenauer [35] provides one of the original works on key predistribution schemes for sensor networks, which is called random key predistribution. Chan [13] extends this scheme to propose a q-composite scheme where q common keys are required to establish a secure link. We note that the Eschenauer scheme is equivalent to the q-composite scheme when $q = 1$. For this chapter, we call the random key predistribution scheme the BASE key predistribution scheme. Additionally, $q = 1$ is solely considered, as the QCOMP link security is significantly degraded for $q > 1$. This is discussed in Chapter 4. and this is justified in subsequent sections when necessary. A unique instance of a network implementing the BASE key predistribution scheme is described by BASE(n, m, P).

We also consider a key predistribution scheme by Du [32] that is based on a key predistribution scheme proposed by Blom [8]. This scheme possesses a λ-secure property, where λ is the threshold for each key. This property requires the adversary to gather $\lambda + 1$ shares of a single polynomial to compromise one key. Delgosha [26] proposed a similar scheme based on multivariate symmetric polynomials representing keys. By requiring $b - 1$ polynomials to be shared in order to establish a secure key, the threshold for each polynomial becomes $\lambda = \binom{t+b-1}{b-1}$ where t is the order of the polynomial. We denote networks using these schemes as TKEY networks, which can be uniquely described by TKEY(n, m, λ, P). If the total memory of the node is m, then $m = \tau(\lambda + 1)$. Each node is preloaded with τ key shares.

Karlof [52] documents several malicious attacks on wireless sensor networks, including spoofed information, selective forwarding, wormhole, sinkhole, and Sybil attacks [30]. This work also establishes the notion of a mote-class attacker and a laptop-class attacker, two distinctions of the level of capability the adversary possess. Additionally, they classify attackers as being *outsiders* or *insiders*. Eavesdropping is one example of an *outsider* attack, while packet dropping is considered to be an *insider* attack. A spoofed node is required for the *insider* attacks. Newsome [71] presents the node-spoofing attack in the context of the

Sybil attack on the random key predistribution scheme. In terms of detecting spoofed nodes, Parno [75] develops line multicast methods to detect the presence of a node in two positions in the network. These methods still require significant overhead to employ. We consider node-spoofing attacks on the key predistribution schemes for wireless sensor networks. With this ability, the adversary is then able to mount attacks of significant threat to the integrity of the network.

6.2.2 Overview of contribution

We examine the property of node-spoofing in wireless sensor networks relative to key predistribution schemes. Currently, the design of security schemes for wireless networks lies in the link security. We investigate node-spoofing as it relates to key predistribution schemes. In this chapter, we present analysis of node-spoofing in sensor networks with respect to the probability of secure key establishment, p_k. This provides a clear and comparable way to assess the quality of the KPS against this attack. We propose variants to the random key predistribution scheme (BASE), which provide improved resilience to node-spoofing attacks on wireless sensor networks. First, we propose regular key predistribution, which enforces a rule that requires each key in the key pool to be stored the same number of nodes. We then introduce the threshold concept to the regular KPS to achieve greater resilience of the KPS against node-spoofing attacks.

6.3 Link and node security in key predistribution schemes

The adversary that we consider in these situations is able to compromise nodes in the network and accumulate the key information of each compromised node. The adversary is then able to launch attacks based on this information. For this chapter, the network is susceptible to link-compromise and node-spoofing attacks based on the fraction of the nodes compromised in the deployed network, p_c. For each node that is compromised, the adversary is able to accumulate all the keys associated with the node.

Probability of link-compromise, p_ℓ: Given that the adversary has captured p_c of the network, it has obtained some fraction P' of the whole key space P. A link k_{ij} is considered to be compromised if the adversary has obtained every key used to establish the link. In

other words, if $k_{ij} \in P'$, then k_{ij} is compromised.

Probability of node-spoof, p_x: With the deployment of the network, nodes are vulnerable to direct compromise from the deployment field. The adversary will redeploy these compromised nodes and use them for malicious purposes. Also, an adversary may compromise nodes and obtain enough key information to present a node identity identical to an already deployed node, which has not been compromised itself. The adversarial model we choose is one that examines the resilience against node-spoofing through indirect means. The nodes that are directly compromised are not considered for a potential node-spoof, but it is assumed that the adversary has recovered all key information inside these nodes. The adversary will attempt to spoof a node which has not been directly compromised.

We consider a node to be successfully spoofed when the adversary selects a particular node identity v'_x and deploys it into an area of the network and 1) passes validation tests from its direct neighbors and 2) establishes a secure link with at least one of its neighbors. A node id $v_{x'}$ is chosen to be spoofed, and the adversary will have t of the m keys in the key ring $K_{x'}$. The spoofed node is detected if the neighborhood $N(v_{x'})$ has any of the $m-t$ keys in $K_{x'}$ the spoofed node does not have. The probability that the spoofed node is able to pass the neighborhood validation test is p_x.

We consider a different node-spoofing adversarial model than compared to [71]. We allow the adversary to attempt to spoof any of the deployed nodes in the network that he has not compromised directly. Newsome [71] allows the adversary to spoof any a node identity that is a combination of m keys for the random key predistribution scheme. In our work, we limit the space of nodes that can be spoofed to those that are deployed.

6.3.1 Adversarial models for the node-spoofing attack

We propose different knowledge models for the adversary with regard to node-spoofing attacks. The idea of the *mote-class* attacker and the *laptop-class* attacker from [52] are considered with these knowledge models for the node-spoofing attack. We define three adversarial models that give the adversary specific capabilities and goals, which determine the subset of nodes that the adversary captures. We assume that the adversary selects its

next node to compromise according to a rule determined by its attack model.

6.3.1.1 Random Attack

We consider the weakest variety of adversarial attack and call it the random attack. In this situation, there is a *mote-class* attacker, whom does not have knowledge of node positions or network deployment topology. Therefore, this attacker is only capable of randomly compromising nodes throughout the network. The next node compromised, v_x, by the adversary according to the random attack model is defined as

$$v_x = v_i, \text{ where } i = \Gamma(n), \tag{6.1}$$

where $\Gamma(n)$ is a uniformly random selection of one of the $n(1-p_c)$ remaining nodes. Nodes are randomly selected from the nodes remaining in the network. After the compromise of a subset of the nodes in the network, the adversary attempts to spoof a random node identity node in a random location of the network. We call the probability of a successful node-spoofing attack using the random attack model to be $p_{x,rand}$.

6.3.1.2 Optimized Attack

We consider a more capable adversary by allowing the attacker to have a global understanding of the topology of the wireless sensor network along with the location and identities of all deployed nodes. We call attacks with this knowledge and capability the optimized attack. In terms of the key predistribution scheme, this attack represents the adversary that goes after the most commonly occurring keys in the network. The compromise of nodes in this way optimally reduces the average number of keys in each node. The optimized attack compromises nodes by removing the maximal number of keys from the network with each successive node capture. The next node to be captured v_x is determined by maximizing the following expression

$$v_x = v_i, \text{ where } i = \arg\max_i \sum_{j=1}^{m} \#\left(k_{v_i}^j\right). \tag{6.2}$$

This adversary is able to achieve a higher average probability of a successful node-spoof than the random attack model, so $p_{x,opt} > p_{x,rand}$.

6.3.1.3 *Identity-optimized Attack*

We consider an attack model that defines a more specific goal for the adversary. We call this attack model the identity-optimized attack. In this situation, the adversary has identified one or several specific node identities to attempt to spoof.

The adversary then compromises nodes, not including those that it wants to spoof, to maximize its chances of a successful spoof of the identified nodes. If the adversary is trying to spoof χ node identities, the rule to determine the next node to compromise is defined by

$$v_x = v_i, \text{ where } i = \arg\max_i \sum_{j=1}^{\chi} |\,(k_j \cap k_i)\,|. \tag{6.3}$$

The difference between the optimized attack and the identity-optimized attack is that the optimized attack is attempting to maximize the average node-spoofing probability for the entire set of deployed nodes. The identity-optimized attack is simply trying to maximize its chances of compromising this small subset of nodes. We expect this variety of attack to be the most threatening to key predistribution schemes applied to sensor networks.

6.4 Node-spoofing and key predistribution schemes

This section presents two key predistribution schemes for use in wireless sensor networks to increase the resilience of a network to the node-spoofing attack. We present the regular key predistribution scheme and the threshold key predistribution scheme, which are two key predistribution schemes that take advantage of a uniform distribution of keys among the nodes in the network. In the original random key predistribution, each key is present in a random number of nodes in the network for any deployment. In a situation where the adversary is able to optimally compromise nodes, BASE is vulnerable. The adversary will be able to select nodes that possess the most frequently occurring keys in the network. We describe both of these key predistribution schemes and then provide comparisons of the random key predistribution to our proposed schemes.

6.4.1 Regular key predistribution

Given a non-uniform distribution of keys in the network, an adversarial attack that is able to compromise the most commonly occurring keys in the network will quickly acquire a

significant portion of the key pool. We propose that every key is required to be present in the same number of nodes in the network. In other words, every key is used the same number of times for a particular network instance. Intuitively, we can see vulnerabilities in key distribution schemes that have a severe non-uniformity in the distribution of keys. An extreme example of this is the use of a global key. Not only will the adversary be able to compromise a significant portion of links by compromising keys that are used most frequently, the adversary will improve its chances of spoofing nodes that are identified by these keys as well. Therefore, by forcing the distribution of the keys in the network to be uniform, this may potentially improve the security against link-compromise and node-spoofing.

We present the regular random key predistribution scheme, which implements this approach. This key predistribution scheme is accomplished and illustrated with a Tanner graph, a graph-theoretic construct used to generate Low-Density Parity-Check (LDPC) codes. Figure 6.1 shows this structure, where the sensor nodes are shown by the vertices $\{v_1, v_2, \ldots, v_n\}$, and the keys are shown by the vertices $\{k_1, k_2, \ldots, k_P\}$. Here, there are m edges incident to each node vertex, x_i, which corresponds to the m keys for each node, x_i. Additionally, the regular key predistribution scheme places a requirement that each key be used an equivalent number of times, a, where $Pa = nm$. In cases where we are given n, m, and P and a is not an integer, we allow keys to be used a and $a+1$ times. The structure of the regular KPS is shown in Figure 6.1, where there are a edges incident to each key vertex k_i and m adjacent to each node vertex v_i.

Similar to REG, a unique instance of the regular key predistribution scheme is defined by $\text{REG}(n, m, P)$. An upper bound for the probability of secure key establishment is

$$p_{k,\text{REG}} \leq \frac{P\binom{a}{2}}{\binom{n}{2}}. \tag{6.4}$$

The probability of secure key establishment for REG compared to BASE is that given equivalent n, m, and P, $p_{k,\text{BASE}} > p_{k,\text{REG}}$.

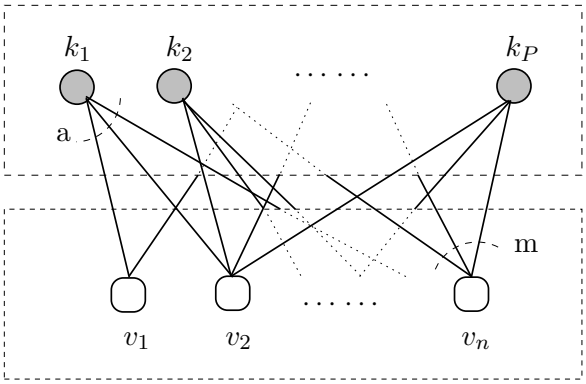


Figure 6.1: Illustration of the key assignment for the regular key predistribution scheme.

6.4.2 Threshold key predistribution

In this section, we attempt to further increase the resilience of key predistribution schemes against node-spoofing attacks by adding threshold security to REG. In this situation, we implement the same Tanner graph-based key assignment to the nodes, but each key is represented by a λ-secure polynomial as used in [26, 32]. We define this threshold key predistribution scheme by TKEY. As mentioned earlier, each node is given key shares from τ polynomials, where $\tau = \lfloor \frac{m}{\lambda+1} \rfloor$ and λ is the threshold. Key shares are assigned to nodes in the TKEY KPS in the same fashion as the polynomials in the multivariate symmetric polynomial key predistribution for $d = 2$. Each key k_i is represented by a two-variable polynomial of dimension λ, $f_i(x_1, x_2) = \sum_{j=0}^{\lambda} \gamma_j x^j y^{\lambda-j}$, where there are a key shares for each key $k_i = \{k_{i(1)}, k_{i(2)}, \ldots k_{i(a)}\}$. For a given key share $k^1_{i(\alpha)}$ in node v_i, the node is given the coefficients to the single variable polynomial, $f_{k_i}(\mathrm{k}^1_{i(\alpha)}, y_2)$. The relationship between the keys, key shares and key polynomials for key k_i is illustrated in 6.2. Any two nodes who have a key share from the same key can establish a secure link by using these key shares to evaluate the corresponding single variable symmetric polynomial, $f_{k_i}(\mathrm{k}_{i(\alpha_1)}, \mathrm{k}_{i(\alpha_2)})$. We note that each stored polynomial occupies $\lambda + 1$ units of memory in each node.

For example, we demonstrate how two nodes v_i, v_j can establish a secure link in a TKEY network. The steps are shown in Table 6.2. Given that the key shares in each node are $(k^1_{i(\alpha_1)}, k^2_{i(\alpha_2)}, \ldots k^\tau_{i(\alpha_\tau)})$ for v_i and $(k^1_{j(\alpha_1)}, k^2_{j(\alpha_2)}, \ldots k^\tau_{j(\alpha_\tau)})$ for v_j, we assume that these two nodes have key shares from the same key, k_ρ, so $k_\rho = k^1_j = k^\tau_i$. Further, in the predistribution

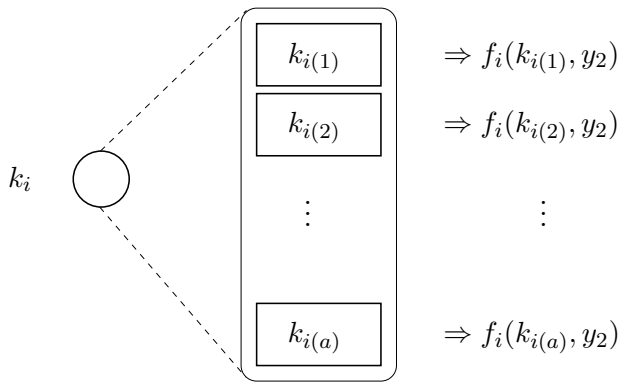


Figure 6.2: Illustration of key k_i and its corresponding key shares and polynomials for the threshold key predistribution scheme.

Table 6.2: Process of establishing a secure link between nodes v_i, v_j in the TKEY key predistribution scheme

node v_i		node v_j
$(k^1_{i(\alpha_1)}, k^2_{i(\alpha_2)}, \ldots k^\tau_{i(\alpha_\tau)})$	$\longrightarrow$	
	$\longleftarrow$	$(k^1_{j(\alpha_1)}, k^2_{j(\alpha_2)}, \ldots k^\tau_{j(\alpha_\tau)})$

Assume $k_\rho = k^1_j = k^\tau_i$.

$$f_{k_\rho}(\mathrm{k}^\tau_{i(\alpha_\tau)}, y_2) = f_{k_\rho}(\mathrm{k}^1_{j(\alpha_1)}, y_2)$$
$$f_{k_\rho}(\mathrm{k}^\tau_{i(\alpha_\tau)}, \mathrm{k}^1_{j(\alpha_1)}) = f_{k_\rho}(\mathrm{k}^1_{j(\alpha_1)}, \mathrm{k}^\tau_{i(\alpha_\tau)})$$

$$k_{ij} = f_{k_\rho}(\mathrm{k}^1_{j(\alpha_1)}, \mathrm{k}^\tau_{i(\alpha_\tau)})$$

phase of TKEY, both of the nodes are preloaded with the coefficients to the single variable polynomial corresponding to the same key, $f_{k_\rho}(\mathrm{k}^1_{i(\alpha_\tau)}, y_2)$ and $f_{k_\rho}(\mathrm{k}^\tau_{j(\alpha_1)}, y_2)$. Because of the symmetric property of the polynomials, nodes v_i, v_j both arrive at the same key by evaluating $f_{k_\rho}(\cdot)$ at the key share from the other node to arrive at the key, $f_{k_\rho}(k^1_{i(\alpha_\tau)}, k^\tau_{j(\alpha_1)})$.

The result is a regular key distribution that has keys with the λ-secure property, TKEY. Although the number of keys in each node is reduced, we expect an increased resilience to node-spoofing attacks with the use of λ-secure polynomials. The probability of secure key establishment is equivalent to the REG scheme, $p_{k,\mathrm{TKEY}} = p_{k(\mathrm{REG})}$. We note that this scheme

is a regular version of [32].

6.5 Evaluation of the regular key predistribution schemes

We now examine the performance of the proposed key predistribution schemes for wireless sensor networks. First, we motivate the proposed key predistribution schemes with a comparison of the distribution of keys between BASE and REG. Second, we examine the link-compromise property of the BASE, REG, and TKEY key predistribution schemes. We see that the uniformity in the distribution of the keys in the network benefits the link-compromise property. Then, we examine the node-spoofing attack on the key predistribution schemes for wireless sensor networks. We consider the node-spoofing attacks from the perspective of the adversarial models described in Section 6.3. We show results comparing the resilience of the node-spoof attack for networks using BASE, REG, and TKEY.

6.5.1 Vulnerability from the distribution of keys

First, we look at the distribution of the keys in BASE. Given the random assignment of keys within the network, the non-uniform distribution of the keys in BASE is illustrated in Figure 6.3. The solid line represents the distribution of the keys in $\text{BASE}(1000, 50, 1000)$ and the dashed line represents $\text{REG}(1000, 50, 1000)$. Although $p_{k,\text{REG}}$ is reduced compared to $p_{k,\text{BASE}}$ for the same P, we see that this property has a direct influence on the probability of link-compromise and node-spoofing. The size of the key pool P required to have p_k for a network of $n = 1000$ and $m = 100$ is shown for BASE and REG in Figure 6.4. the difference in the key pool size increases as p_k decreases. In fact, $\text{BASE}(1000, 100, 100000)$ and $\text{REG}(1000, 100, 50000)$ both have $p_k = 0.10$, where there is a much smaller discrepancy between networks with $p_k = 0.78$ in $\text{BASE}(1000, 100, 7000)$ and $\text{REG}(1000, 100, 6250)$.

The non-uniform distribution of keys in the network allows the adversary to compromise the keys that occur the most number of times in the network. This affects both the probability of node-spoofing as each key is used a uniform number of times in the network. The adversary is initially unable to mount an optimal attack on the distribution of keys.

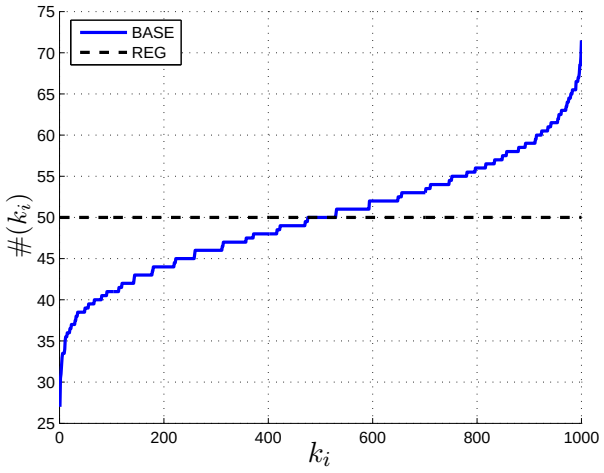


Figure 6.3: Comparison of the distribution of key usage between REG and BASE with $n = 1000$, $m = 50$, and $P = 1000$.

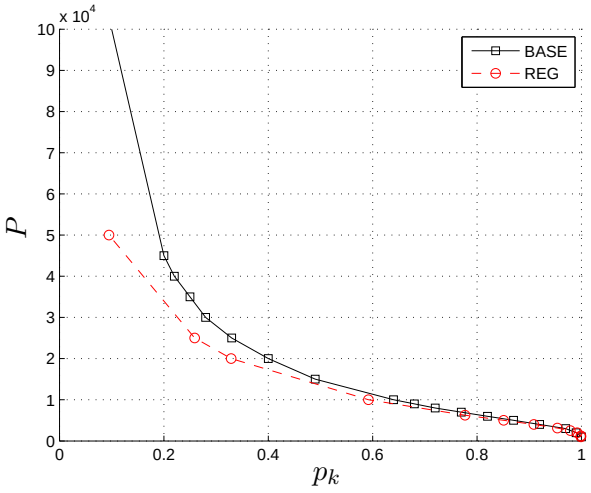


Figure 6.4: Comparison of the size of the key pool, P versus the probability of secure key establishment REG and BASE with $n = 1000$, $m = 100$.

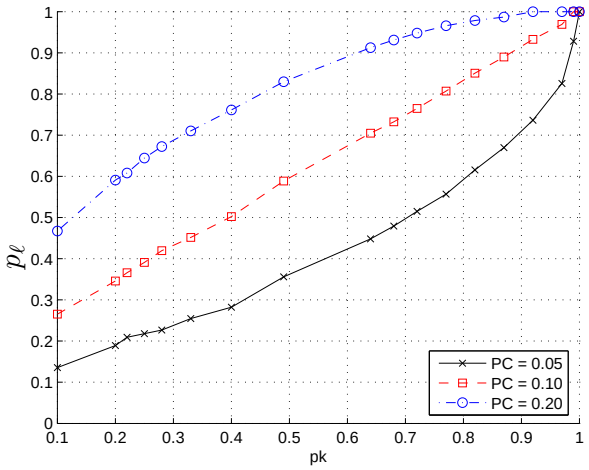


Figure 6.5: p_ℓ vs. p_k for random key predistribution given a random node-compromise for BASE$(1000, 100, P)$

6.5.2 Relating link-compromise and node-spoofing attacks

In our analysis of the node-spoofing attack, we use the probability of link-compromise p_ℓ to determine the range of p_k that the node-spoofing attacks will be considered. When looking at p_ℓ for the random key predistribution scheme, networks with lower p_k exhibit greater resilience to link compromise. This property is illustrated in Figure 6.5, where p_ℓ vs. p_k is plotted for BASE$(1000, 100, P)$, with P varied to obtain the range of p_k. The figure shows the plots p_ℓ for $p_c = \{0.05, 0.10, 0.20\}$. This plot suggests that considering the lower range of p_k offers more resilience to link-compromise attacks. For our investigation into node-spoofing, we consider instances of KPS for $0.10 < p_k < 0.20$. This way we can examine networks against node-spoofing while the link security is also kept high.

First, we compare the probability of link-compromise of the basic scheme BASE with both of the regular key predistribution schemes, REG and TKEY. As shown in Figure 6.6, we plot the probability of link-compromise against the probability of secure key establishment for a network of $n = 1000$ and $m = 100$. The networks that we have shown in this plot are BASE$(1000, 100, 100000)$, REG$(1000, 100, 50000)$, TKEY$(1000, 1, 100, 2, 8000)$, and TKEY$(1000, 1, 100, 9, 900)$. The parameters for each of these key predistribution schemes result in a network with $p_k = 0.10$. We have shown the probability of link-compromise, p_ℓ,

given a random node-compromise attack model. This shows that REG suffers a slight decrease in resilience to link-compromise compared to BASE. We also see that the networks using TKEY have an increased resilience to the link-compromise attack compared to BASE. This behavior is attributed to the threshold property of the keys.

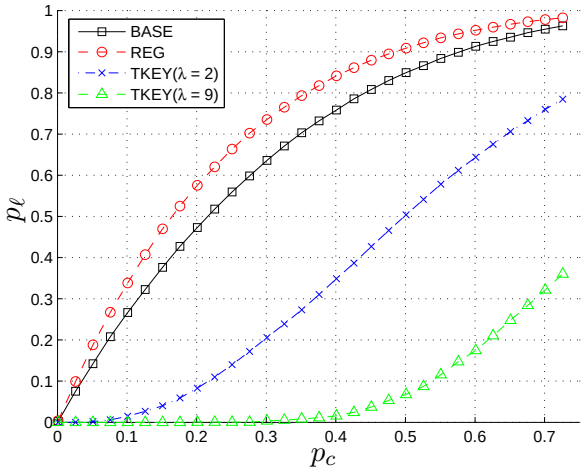


Figure 6.6: p_ℓ vs. p_c given a random node-compromise attack for BASE, REG, and TKEY($\lambda = [2, 9]$), where $p_k = 0.10$.

As described in Section 6.3 for the node-spoofing attack, we can define an optimal attack with respect to the link-compromise attack. In this case, the adversary attempts to maximize the number of links it compromises with each successive node capture. Therefore, the rule to determine the next node to compromise for the link-optimized attack model is defined by

$$v_x = v_i, \text{ where } i = \arg\max_i \sum_{j=1}^{m} \# \left(k_i^j \right). \tag{6.5}$$

The link-optimal attack is illustrated in Figure 6.7, where the random attack and the link-optimized attack, in addition to the gain realized from the optimized attack, are shown. The gain is almost 20% at $p_c = 0.15$. This optimal attack is revisited later in this chapter.

6.5.3 Regular key predistribution schemes and node-spoofing attacks

To illustrate the difference in node-spoofing resilience between these key predistribution schemes, we consider the network where $n = 1000$, $m \leq 100$. We present simulation results

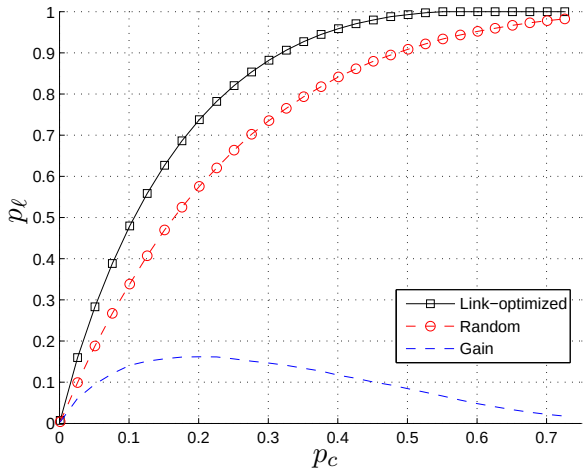


Figure 6.7: p_ℓ vs. p_c given a random and link-optimized node-compromise attack for REG. Gain from the link-optimized attack is also shown.

of node-spoofing in wireless sensor networks as a function of p_c. In the simulations, we have implemented REG and BASE, deployed a set of n nodes and removed $p_c n$ nodes according to either the random or optimized node-compromise attack model. The communication radius is set to $r = 0.15$, which is sufficient for the network to have global connectivity. As studied in Chapter 4, the communication range required for connectivity is dependent upon the network size and the probability of secure key establishment. Additionally, the communication range impacts p_x in that the expected neighborhood size or the number of validating nodes is equal to $|N_x| = \pi r^2 n$. Additionally, we compare different KPS where p_k and m are the same, allowing for a fair comparison in terms of initial network connectivity and memory usage. We now look at the node-spoofing attack while considering the adversarial models that Section 6.3 describes: the random, optimized, and the identity-optimized attacks.

6.5.3.1 Analysis of the node-spoof attack on wireless sensor networks and key predistribution schemes

Determining the probability of node-spoof, p_x, and link-compromise, p_ℓ, for REG and TKEY can be derived with the use of the classic balls and bins problem [95]. In the original problem, there are N balls independently thrown into M bins. This problem can be adapted

to determining p_x and p_ℓ for the regular key predistribution schemes by having one bin per key and the balls as the key shares present in the nodes. The balls represent the keys or key shares that are compromised by the adversary. Each bin has a finite capacity of a, and in our situation, there are exactly a balls of each key randomly distributed in the nodes. This interpretation can be illustrated with Figure 6.1.

In this section, we determine the average probability for link-compromise and node-spoof for the random node-compromise adversarial attack model. For REG, we consider P bins, all with a capacity of a. Given p_c, there are $p_c nm$ balls distributed into the bins. We want to determine the expected number of bins with at least one ball in it, given p_c. The fraction of bins with a ball in it is interpreted to be P', the size of the compromised key pool. Similarly for TKEY, we consider P bins with capacity a, and there are $p_c n\tau$ balls distributed into the bins. We want to determine the expected number of bins with at least $\lambda+1$ balls. Here, the fraction of bins with $\lambda+1$ balls in it is interpreted to be P', the number of compromised polynomials. The derivation for these expressions is found in Appendix B. With this expression, the probability of node-spoof can also be derived. The probability of link-compromise for REG and TKEY are defined by

$$p_{\ell\varphi}(\psi,\varphi) = \frac{\sum_{k=\varphi}^{a}\sum_{j=0}^{P-1}(-1)^j\binom{P-1}{j}\binom{P-2+(\psi-k-j(a+1))}{(\psi-k-j(a+1))}}{P\sum_{\ell=0}^{P}(-1)^\ell\binom{P}{\ell}\binom{P-1+(\psi-\ell(a+1))}{(\psi-\ell(a+1))}}, \tag{6.6}$$

where $p_{\ell,\text{REG}} = p_{\ell\varphi}(p_c nm, 1)$ and $p_{\ell,\text{TKEY}} = p_{\ell\varphi}(p_c n\tau, \lambda+1)$.

For the probability of node-spoof, p_x, we use the result of p_ℓ. The success of the node-spoofing attack is discussed in 6.3. First, we determine the probability that the adversary has compromised i of the m keys for the node, k_{v_i}, that it is attempting to spoof (for TKEY, we substitute τ for m). We define this as

$$Pr(|k_{v_i} \cap P'| = i) = \binom{m}{i}(p_x)^i(1-p_x)^{m-i}. \tag{6.7}$$

Second, we determine the probability that the spoofed node passes the validation step, given that the adversary has i of the m keys of the adversary. The validation test is passed if the neighborhood does not possess any of the $m-i$ keys that the adversary does not have for the node, k_{v_i}. The keys in the neighborhood is defined to be $P_{N_{v_i}}$, and the keys that

the adversary does not have is the complement of the compromised key pool, $(P')^c$. The probability that the spoofed node passes the validation test from the neighbors is bounded by

$$Pr([(P')^c \cap k_{v_i}] \cap P_{N(v_i)} = \varnothing \mid \ |k_{v_i} \cap P'| = i) \leq \left(\frac{\pi r^2 (a-1)(m-i)}{n}\right). \tag{6.8}$$

From (6.7) and (6.8), we can determine the probability of a successful node-spoof attack for the regular key predistribution schemes. An upper-bound for this probability can be described by the following

$$\begin{aligned} p_x &= Pr(|k_{v_i} \cap P'| = i) Pr([(P')^c \cap k_{v_i}] \cap P_{N(v_i)} = \varnothing \mid \ |k_{v_i} \cap P'| = i) \\ p_x &\leq \sum_{i=1}^{m} \binom{m}{i} (p_{\ell\varphi}(\psi, \varphi))^i (1 - p_{\ell\varphi}(\psi, \varphi))^{m-i} \left(\frac{\pi r^2 (a-1)(m-i)}{n}\right). \end{aligned} \tag{6.9}$$

6.5.3.2 Results for node-spoofing attacks with a random/optimized adversary

When considering the random and optimized adversarial node-compromise models, the probability of a successful node-spoofing attack is interpreted to be the average probability of an adversary to spoof a node in a random location in the network. We consider both the random node-compromise attack model and the optimized node-spoofing attack.

Figures 6.8 and 6.9 show the fraction of the network compromised p_c required for the adversary to have a probability of a successful node-spoof of $p_x = 0.10$ for values of the probability of secure key establishment, $0.10 < p_k < 0.25$. Figure 6.8 depicts the node-spoofing attack for the random adversarial attack model, and Figure 6.9 shows the same for the optimized node-compromise adversarial model. We note that the gain of the adversary by being able to optimally compromise nodes for the sake of the node-spoofing attack is nearly 5% for BASE and REG, and 10% for both TKEY networks. We interpret the gain to be the reduction in the percentage of the network the adversary has to compromise to achieve $p_x = 0.10$.

When comparing the resilience of the node-spoofing attack for BASE and REG, the simulation results show that BASE has a higher vulnerability than REG for both of the adversarial

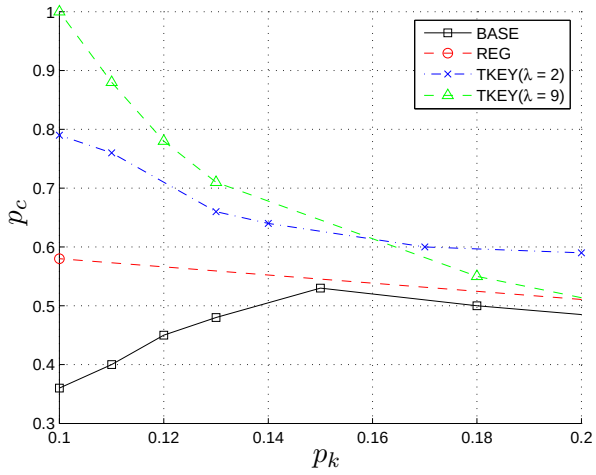


Figure 6.8: p_c vs. p_k for the adversary to have a 10% chance of a successful node-spoof given a random node-compromise adversarial model for networks with parameters $n = 1000$ and $m = 100$ and using BASE, REG, and TKEY$[\lambda = 2, 9]$.

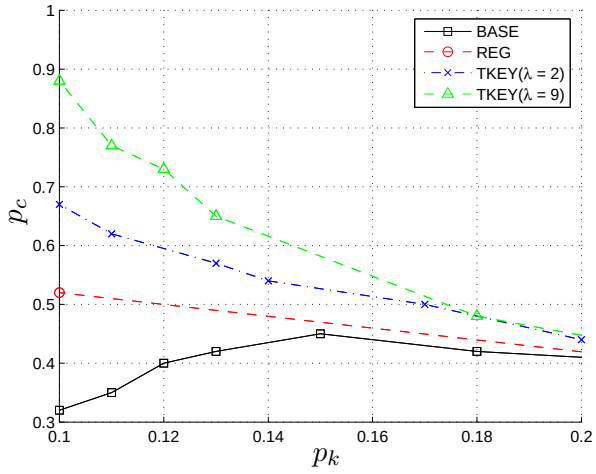


Figure 6.9: p_c vs. p_k for the adversary to have a 10% chance of a successful node-spoof given an optimized node-compromise adversarial model for networks with parameters $n = 1000$ and $m = 100$ and using BASE, REG, and TKEY$[\lambda = 2, 9]$.

models. When $p_k = 0.10$, REG requires 25% more nodes to be compromised than BASE to achieve $p_x = 0.10$. These figures also show the increased resilience to the node-spoofing attack of TKEY compared to both REG and BASE. For $p_k = 0.10$, TKEY($\lambda = 9$) requires 0.40 more of the network to be compromised compared to BASE to achieve $p_x = 0.10$. This result is attributed to the λ-secure property for each key used in TKEY. Additionally, the plots of p_x versus p_c for $p_k = 0.10$ are shown in Figure 6.10, for the random attack model, and in Figure 6.11, for the optimized attack model. These results correspond to the previous discussion that examined the fraction of the network required to be compromised in order for the adversary to have a certain probability of spoof a node successfully. The plots for are shown for the four instances of key predistribution schemes: BASE, REG, and TKEY($\lambda = [2, 9]$).

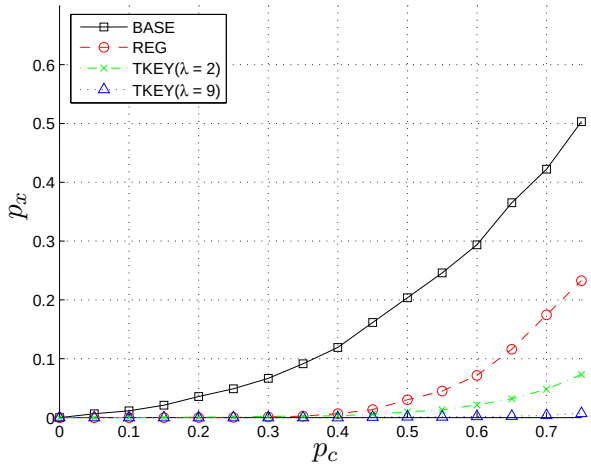


Figure 6.10: p_x vs. p_c given random node compromises for BASE, REG, and TKEY$[\lambda = 2, 9]$ for $n = 1000$ and $m = 100$.

6.5.3.3 Maximizing λ for TKEY

In terms of TKEY, we are able to adjust the λ parameter while keeping p_k and m the same. We can adjust this parameter to optimize the resilience of the network against the node-spoofing attack. We optimize this parameter by determining the value of λ that results in the adversary to compromise the greatest number of nodes in the networking scenario we have been considering. For the optimal node-compromise attack, we examine the required

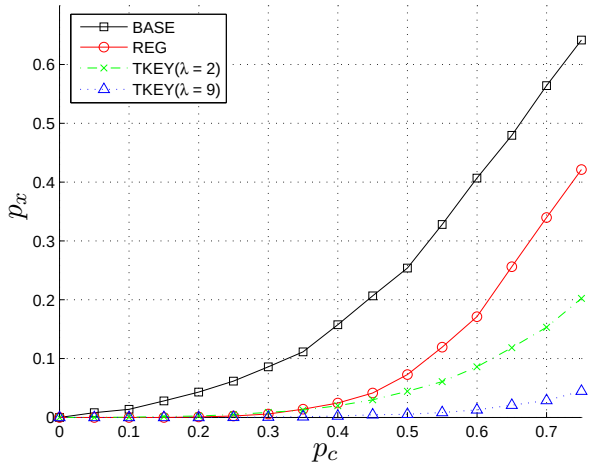


Figure 6.11: p_x vs. p_c given optimized node compromises for BASE, REG, and TKEY$[\lambda = 2, 9]$ for $n = 1000$ and $m = 100$.

fraction p_c of the network required to give the adversary a 10% chance to successfully spoof a node as a function of the threshold parameter, λ. We call the value of λ that maximizes the network against the node-spoofing attack to be λ_{MAX}. For $n = 1000$, $m = 100$, and $p_x = 0.10$, the network is most resilient to the node-spoofing attack with $\lambda_{MAX} = 9$ This result is shown in Figure 6.12. The value of λ_{MAX} depends on the network parameters and the probability of successful node-spoof for which it is defined. For example, given the initial scenario, if the TKEY key predistribution parameters are chosen with $p_x = 0.25$, choosing $\lambda_{MAX} = 9$ may result in the network providing sub-optimal resilience against the node-spoof attack. This sub-optimal behavior is due to the threshold behavior of $p_{x,\text{TKEY}}$.

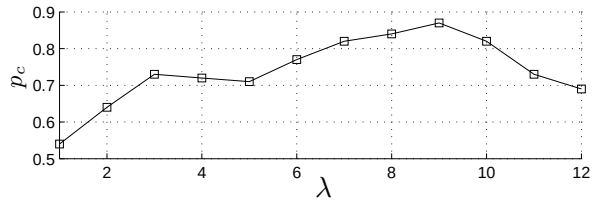


Figure 6.12: p_c vs. λ for an optimized node-compromise adversarial model.

6.5.3.4 Results for node-spoofing attacks with an identity-optimized adversary

We also consider the identity-optimized adversarial model. This attack establishes a scenario where the adversary is attempting to spoof χ specific node identities. The adversary may have identified a specific set of nodes that it aims to control. These nodes may be attractive nodes in the routing dynamics, meaning that these nodes are close to the base station, and nodes in the network are attempting to route their packets to these nodes as fast as possible. Additionally, successful spoofing of these nodes may allow the adversary to alter the routes in the network advertising the presence of these nodes in other locations of the network. This results in packets being routed through and to suboptimal routes and locations. This decreases packet reliability and increases packet delivery rates. Further, more sophisticated attacks such as sinkhole or blackhole attacks could be implemented as well.

We look at the identity-optimized attack for BASE, REG, and TKEY$[2, 9]$ and determine the average fraction of the network required to be compromised in order to spoof the targeted nodes. The spoofed nodes are placed into a random location in the network. We note that given location information of the neighborhood may give rise to another optimal attack for wireless sensor networks with key predistribution schemes. However, we assume that the adversary is not aware of its neighborhood. We have generated results for the identity-optimized adversarial attack through simulation. The result of the identity-optimized spoofing attack is that the success of the spoofing occurs at a sharp threshold. We note that the number of nodes required to successfully spoof one specific node identity ($\chi = 1$) for each of the key predistribution schemes BASE, REG, and TKEY$[2, 9]$, were all very similar. This is shown in Figure 6.13 for BASE and REG.

We also examine the identity-optimized adversarial attack for $\chi = 1$ to 5 nodes, comparing random key predistribution to the regular key predistribution schemes. In Figure 6.13, we show p_x vs. p_c for BASE$(1000, 100, 100000)$ and REG$(1000, 100, 50000)$ with $\chi = 1$ and $\chi = 5$. Here, REG requires nearly twice the number of nodes to be compromised compared to BASE for the identity-optimized adversarial attack of $\chi = \{1, 5\}$ nodes. In Figure 6.14, we see the fraction p_c of the network required to be compromised by the identity-optimized attack to achieve a successful node-spoof for each KPS. Our results show that

REG, TKEY($\lambda = 2$)and TKEY($\lambda = 9$) possess similar behavior with regard to this attack for every value of χ. BASE performs the worst with regard to this attack, where for $\chi = 5$, the BASE network requires $p_c = 0.25$, where as each of the regular KPS require between $pc = [0.35, 0.40]$. This disparity is significant. The reduced resilience of this attack is caused by the large key pool size. In In REG and TKEY, the verifying (neighbor) nodes have a much larger fraction of the key pool, significantly reducing the number of partially spoofed nodes that pass the validation test.

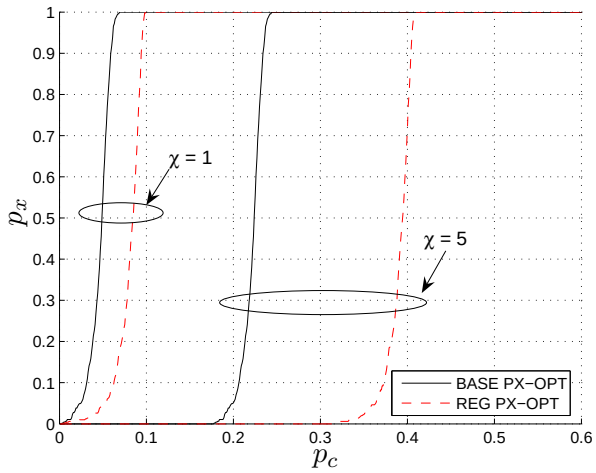


Figure 6.13: p_x vs.p_c given the identity-optimized attack model, $\chi = \{1, 5\}$, for BASE, REG, $n = 1000$, $m = 100$, $p_k = 0.10$.

6.6 Summary

We have examined the node-spoofing attack for key predistribution schemes used in wireless sensor networks. Additionally, we have proposed two key predistribution schemes that provide a higher resilience to the node-spoofing attack: regular key predistribution, REG, and threshold regular key predistribution scheme, TKEY. Both of these schemes provide an increased resilience to the node-spoof attack in the lower range of p_k, where link security is high. The gains are realized by enforcing a uniform distribution of keys present in the nodes in the network and by implementing a λ-secure property for each of the keys. TKEY provides resilience against the node-spoof attack to a certain threshold, where beyond the threshold, the key predistribution scheme is vulnerable. Given a situation where the expected number

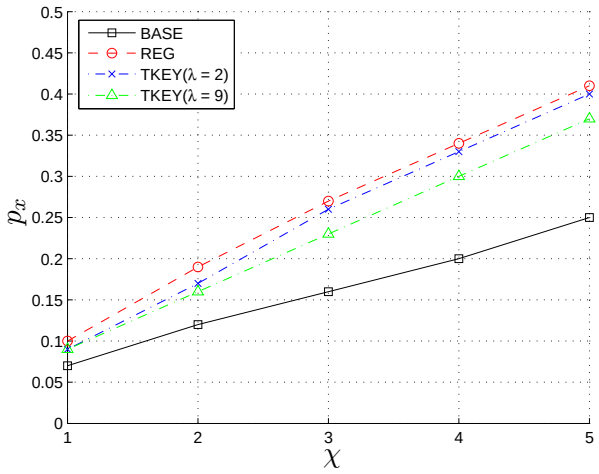


Figure 6.14: p_x vs.p_c given the identity-optimized attack model, $\chi = \{1, 5\}$, for BASE, REG, $n = 1000$, $m = 100$, $p_k = 0.10$.

of compromised nodes is bounded, TKEY is more desirable for use over REG or other key predistribution schemes. In conclusion, we infer that adopting a uniform distribution of the usage of keys in the network can improve the resilience of node-spoofing spoofing attacks for other key predistribution schemes.

CHAPTER VII

CONCLUSION

To conclude this thesis, we provide a summary of the contributions to the research areas of cryptography, network security, and wireless sensor networks. This thesis has documented an analysis of link and network security pertaining to resource-constrained wireless devices. With constraints on available energy, processing capability, transmission power, and memory, new algorithms and schemes need to be developed and analyzed for proper deployment and usage. Furthermore, the operation of these networks and their constituent devices occurs in harsh and also adversarial environments, where security services are necessary. The ability to perform intended tasks with the added requirement of information security exacerbates the constraints of these devices. We have developed and analyzed schemes for these networks of resource-constrained devices to achieve sufficient resilience against adversarial attacks while still providing adequate levels of networking performance. We have separated the research of this thesis into contributions of link security and network security.

In Chapter 3, we analyzed the security of a new cryptographic primitive for use in handheld devices [16, 17, 24]. The specific contributions with regard to link security that were presented in this thesis are as follows.

1. We have analyzed the security of a private key cryptosystem based on the finite-field wavelet, the Wavelet Block Cipher (WBC). We investigated classical and new cryptanalytic attacks against the wavelet cryptosystem. We did not find any vulnerabilities of WBC with regard to classical cryptanalytic techniques. Furthermore, we developed variants of the interpolation, delta function, and discrete Fourier transform-based attacks and showed that these attacks cannot do better than the exhaustive key search method.

2. We studied the computational complexity of WBC, the polyphase representation of the wavelet transform allows for WBC decryption to have almost half the complexity

compared to WBC encryption. The 128-bit WBC encryption has a computational complexity lower than DES and comparable to AES with key sizes of 64 and 128 bits, respectively. Additionally, the complexity of the wavelet decryption is lower than both DES and AES decryption.

We considered several problems within the topic of wireless sensor networks [19, 23]. The underlying research problem that we analyzed was the performance of networks in the presence of an adversarial attack. We considered key predistribution schemes for use with wireless sensor networks and studied the effect of node-compromise attacks. In Chapter 4, we considered the resilience of global network connectivity in the context of node-compromise attacks [14, 15, 79, 80]. The contributions pertaining to secure connectivity and node-compromise attacks are as follows.

1. We established expressions for the communication range required for connectivity in several different wireless network models employing key predistribution. We determined the resource cost of implementing key predistribution schemes on wireless sensor networks in relation to global network connectivity.

2. We examined the effect of node-compromise attacks on the communication range required for networks to maintain secure connectivity. Determining the effect of node-compromise attacks on these networks required the derivation of the rate of link compromise for key predistribution schemes.

3. We proposed a resiliency-connectivity metric, which is a measure of communication range as a function of the magnitude of the adversarial node-compromise attack. This metric allows for networks to compare their resilience in terms of connectivity against node-compromise attacks.

4. Our analysis shows that the multivariate symmetric polynomial key predistribution scheme possesses greater resilience, in terms of network connectivity, to node-compromise attacks compared to the random key predistribution scheme.

As an extension to the work presented in Chapter 4, we investigated other network properties and their resilience to node-compromise attacks. In Chapter 5, we considered the metrics of packet latency and achievable throughput for wireless sensor networks [96]. Our contributions are as follows.

1. We investigated the resilience of average packet latency as a function of the number of nodes compromised. We considered different key predistribution schemes and compared their resilience to node-compromise attacks with regard to several network properties.

2. We determined the maximum achievable throughput of a network for various degrees of node-compromise attacks. Similar to the analysis of packet latency, we studied the resilience of the achievable throughput of a network employing key predistribution schemes as a function of the number of compromised nodes.

3. To present another perspective of the latency within networks employing key predistribution schemes, we considered the distance from the sink node as a parameter. The average packet latency of nodes at various distances from the sink node was considered.

4. We established the connection between the resilience results of maximum throughput and average packet latency and the results shown for connectivity in Chapter 4 for the multivariate symmetric key predistribution and random key predistribution schemes.

We also consider the node-spoof attack on wireless sensor networks [18]. Given a successfully spoofed node in the network, the adversary can launch more powerful attacks. We studied the node-spoof attack and presented approaches to increase the resilience of a network to this variety of attack.

1. We developed an adversarial model for use with the node-spoof attack. We studied the probability of node-spoof as a function the probability of secure key establishment for networks employing various existing key predistribution schemes.

2. We proposed a regular key predistribution scheme to provide increased resilience of networks to node-spoof attacks. Our design approach fixed the number of times each key appeared in a node in the network. We showed that this method improved the resilience of the random key predistribution scheme against node-spoof attacks.

3. We also proposed a scheme that establishes a threshold effect on each of the keys. It was shown that the threshold regular key predistribution scheme provided greater resilience to spoofing attacks compared to both the random and the regular key predistribution schemes.

4. We considered optimal node-compromise attacks in terms of maximizing the probabilities of link-compromise and node-spoofing. When optimizing for the node-spoofing attack, it was shown that it provides improvement to the probability of link-compromise compared to a random node-compromise approach. Optimizing for link-compromise does not demonstrate this property. In some cases, a link-compromise optimizing attack fares only as good as the random node-compromise attack for node-spoofing. It is determined that the node-spoofing optimal attack is more powerful than the link-compromise optimal attack.

APPENDIX A

MATRIX REPRESENTATION OF THE ELEMENTARY ENCRYPTION BLOCKS FOR THE WAVELET TRANSFORM

In this appendix, we define the matrix operations for which the elementary encryption block for the wavelet transform can be represented. The output $y(n)$ of the wavelet transform can be represented with matrix operations as a function of $x(n)$ and the secret key or filter coefficients, $e_{00}(n)$. Define the following matrices:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix},$$

where A represents the downsampling function $[\downarrow 2]$, and B is a unit shift to the left that is followed by the downsampling function, $[\downarrow 2]$. Using (2.4), it can be shown from Figure 3.2 that the input and output relation of the elementary encryption block can be written as $y = (G_0 A + G_1 B)x$, where the input and output are represented by two vectors x and y,

respectively. Equivalently, in the polyphase representation, we define

$$C = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \vdots & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$$

where C is the upsampling function $[\uparrow 2]$, D represents the upsampling function $[\uparrow 2]$ followed by a unit shift to the right, and E_{oo} is the matrix representation of the polyphase filter. The matrix S represents the $z^{-\frac{M-1}{2}}$ cyclic shift operation. In the matrix S, the 1 in the first row is at the $(\frac{M-1}{2}^{th})$ column. Therefore, the input and output relation of the elementary encryption block in Fig. 3.3 can be written as

$$Y = (DE_{00}A + CE_{00}B + C(E_{00} + S)A + D(E_{00} + S)B)x \tag{A.1}$$

$$Y = Tx. \tag{A.2}$$

We will refer to the elementary encryption block by the matrix T. Similar matrix representation exists for the decryption block of Fig. 3.5. We denote this matrix as F for the remainder of this work.

APPENDIX B

PROBABILITY OF LINK COMPROMISE DERIVATION FOR REGULAR AND REGULAR THRESHOLD KEY PREDISTRIBUTION

In this appendix, we derive the probability of link compromise based on the classical balls and bins problem. First, we state some well-known combinatorial relationships. Then, we derive expressions for $p_{x,REG}$ and $p_{x,TKEY}$.

B.1 Facts

We use these combinatorial relationships for our derivation.

$$i\binom{L}{i} = \frac{iL!}{i!(L-i)!} = L\binom{L-1}{i-1} \tag{B.1}$$

$$(1 + x + x^2 + \cdots + x^m)^L = \frac{1 - x^m}{(1-x)^L} \tag{B.2}$$

$$\frac{1}{(1-x)^L} = \sum_k \binom{L-1+k}{k} x^k \tag{B.3}$$

$$(1 - x^m)^L = \sum_{j=0}^{L} (-1)^j \binom{L-1}{j} x^{jm} \tag{B.4}$$

B.2 Probability of link-compromise

We use the balls and bins problem to derive expressions for the probability of link compromise for the regular and regular threshold key predistribution schemes. We consider P keys (bins) or polynomials and ψ total key shares (balls). Each key is present a times, and φ key shares are required for a polynomial to be compromised. In the case of REG, $\varphi = 1$.

We find the total combinations of n compromised key shares distributed among L polynomials with a maximum m key shares per polynomial. We want the coefficient of $A_\psi x^\psi$

for the following expression

$$\sum_{i=0}^{P} i\binom{P}{i}(x^{\varphi}+x^{\varphi+1}+\cdots x^{a})^{i}(1+x+\cdots x^{\varphi-1})^{P-i} \tag{B.5}$$

$$P(1+x+\cdots x^{\varphi-1})\sum_{i=1}^{P}\binom{P-1}{i-1}(x^{\varphi}+x^{\varphi+1}+\cdots x^{a})^{i}(1+x+\cdots x^{\varphi-1})^{(P-i-1)}$$

$$P(x^{\varphi}+x^{\varphi+1}+\cdots x^{a})(1+x+\cdots+x^{a})^{P-1}$$

$$P(x^{\varphi}+x^{\varphi+1}+\cdots x^{a})\left(\frac{1-x^{m+1}}{(1-x)}\right)^{P-1}$$

$$P(x^{\varphi}+x^{\varphi+1}+\cdots x^{a})\left(1-x^{a+1}\right)^{P-1}\frac{1}{(1-x)^{P-1}}$$

$$P(x^{\varphi}+x^{\varphi+1}+\cdots x^{a})\left(\sum_{\ell=0}^{\infty}\binom{P-2+\ell}{\ell}x^{\ell}\right)\left(\sum_{j=0}^{P-1}(-1)^{j}\binom{P-1}{j}x^{(a+1)j}\right) \tag{B.6}$$

In (B.6), the $(x^{\varphi}+x^{\varphi+1}+\cdots x^{a})$ term provides degrees φ through a. The second and third terms need to provide degrees $\psi-\varphi$ through $\psi-a$. The third term provides degrees of x in multiples of $(a+1)j$. The $A_{\psi}x^{\psi}$ term is:

$$x^{\psi}\sum_{k=\varphi}^{a}\sum_{j=0}^{P-1}(-1)^{j}\binom{P-1}{j}x^{j(a+1)}\binom{P-2+(\psi-k-j(a+1))}{(\psi-k-j(a+1))} \tag{B.7}$$

We also find the total number of combinations of ψ key shares distributed to P polynomials with a maximum of a per bin. This is a known result from [95]. This is obtained by finding the x^{ψ} coefficient in $(1 + x + \cdots x^a)^P$. This is determined by

$$(1 + x + \cdots x^a)^P = \left(\frac{1 - x^{a+1}}{1 - x}\right)$$
$$\left(\sum_{\ell=0}^{\infty} \binom{P - 1 + \ell}{\ell} x^{\ell}\right) \left(\sum_{j=0}^{P} (-1)^j \binom{P}{j} x^{(a+1)j}\right)$$

Similar to the previous derivation, the second term provides the degrees of x in multiples of $(a + 1)j$. The x^{ψ} term will be:

$$x^n \sum_{j=0}^{P} (-1)^j \binom{P}{j} \binom{P - 1 + (\psi - j(a + 1))}{(\psi - j(a + 1))} \tag{B.8}$$

Therefore, the general expression for the probability of link compromise is determined from (B.6) and (B.8). This expression is a function of φ, the number of key shares per key required for any key to be compromised.

$$p_{\ell\varphi}(\psi, \varphi) = \frac{\sum_{k=\varphi}^{a} \sum_{j=0}^{P-1} (-1)^j \binom{P-1}{j} \binom{P-2+(\psi-k-j(a+1))}{(\psi-k-j(a+1))}}{P \sum_{\ell=0}^{P} (-1)^{\ell} \binom{P}{\ell} \binom{P-1+(\psi-\ell(a+1))}{(\psi-\ell(a+1))}} \tag{B.9}$$

Finally, the probability of link compromise for REG and $TKEY$ are as follows

$$p_{\ell,REG} = p_{\ell\varphi}(p_c n m, 1) \tag{B.10}$$

and

$$p_{\ell,TKEY} = p_{\ell\varphi}(p_c n \tau, \lambda + 1). \tag{B.11}$$

These relationships are also used to determine the probability of a successful node-spoofing attack for regular key predistribution schemes.

REFERENCES

[1] Adams, C., "Constructing symmetric ciphers using the cast design procedure," *Designs, Codes, and Cryptography*, vol. 12, no. 3, pp. 283–316, 1997.

[2] Akkaya, K. and Younis, M., "A survey of routing protocols in wireless sensor networks," *Elsevier Ad Hoc Network Journal*, vol. 3, no. 3, pp. 325–349, 2005.

[3] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., "A survey on sensor networks," *IEEE Communications Magazine*, pp. 102–114, August 2002.

[4] Biham, E., "Cryptanalysis of multiple modes of operation," *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, vol. 11, pp. 45–58, Winter 1998.

[5] Biham, E., "Cryptanalysis of triple modes of operation," *Journal of Cryptology: the journal of the International Association for Cryptologic Research*, vol. 12, no. 3, pp. 161–184, 1999.

[6] Biham, E. and Shamir, A., *Differential Cryptanalysis of the Data Encryption Standard*. No. 1, Springer-Verlag, 1993.

[7] Blahut, R., *Algebraic Methods for signal processing and communications coding*. Springer-Verlag, 1991.

[8] Blom, R., "An optimal class of symmetric key generation systems," *Advances in Cryptology: Proceedings of EUROCRYPT 84*, no. 209, pp. 335–338, 1984.

[9] Bollobás, B., *Random Graphs*. Cambridge University Press, second ed., 2001.

[10] Brown, L., Pieprzyk, J., and Seberry, J., "LOKI - a cryptographic primitive for authentication and secrecy applications," *Advances in Cryptology - Auscrypt'90*, vol. LNCS 453, pp. 229–236, 1990.

[11] Carman, D., Kruus, P., and Matt, B., "Constraints and approaches for distributed sensor network security," Tech. Rep. 00-010, NAI Labs, 2000.

[12] Chan, H. and Perrig, A., "Pike: peer intermediaries for key establishment in sensor networks," *Proceedings IEEE of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, vol. 1, pp. 524–535, March 2005.

[13] Chan, H., Perrig, A., and Song, D., "Random key predistribution schemes for sensor networks," *2003 IEEE Symposium on Security and Privacy*, p. 197, 2003.

[14] Chan, K. and Fekri, F., "Node compromise attacks and secure connectivity," *SPIE Defense and Security Symposium, 6758-33*, April 2007.

[15] Chan, K. and Fekri, F., "A resiliency-connectivity metric in wireless sensor networks with key predistribution schemes and node compromise attacks," *to appear in Elsevier Physical Communication*, March 2008.

[16] Chan, K. S. and Fekri, F., "On the security of the finite-field wavelet-based block cipher," *in Proc. of IASTED International Conference on Communication, Network, and Information Security (CNIS 2003)*, pp. 91–96, December 2003.

[17] Chan, K. S. and Fekri, F., "A block cipher cryptosystem using wavelet transforms over finite fields," *IEEE Transactions on Signal Processing*, vol. 52, pp. 2975–2991, October 2004.

[18] Chan, K. and Fekri, F., "Resisting node spoofing attacks in random key predistribution schemes: A uniform design," *submitted to MILCOM 2008*, November 2008.

[19] Chan, K., Pishro-Nik, H., and Fekri, F., "Analysis of hierarchical routing protocols for wireless sensor networks," *in the Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2005)*, March 2005.

[20] Courtois, N. T. and Pieprzyk, J., "Cryptanalysis of block ciphers with overdefined systems of equations," *ASIACRYPT02*, pp. 267–280, 2002.

[21] Courtois, N. T. and Pieprzyk, J., "Cryptanalysis of block ciphers with overdefined systems of equations," *ASIACRYPT02*, pp. 267–280, 2002.

[22] Daemen, J. and Rijmen, V., *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer Verlag, 2002.

[23] Delgosha, F., Ayday, E., Chan, K., and Fekri, F., "Security services for wireless sensor networks using sparse random coding," *Third Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks*, October 2006.

[24] Delgosha, F., Chan, K., and Fekri, F., "Multivariate symmetric cryptography using finite-field wavelets," *2007 Hawaii and SITA Joint Conference on Information Theory, HISC2007*, May 2007.

[25] Delgosha, F. and Fekri, F., "Stream cipher using finite-field wavelets," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp. 689–692, March 2005.

[26] Delgosha, F. and Fekri, F., "Threshold key-establishment in distributed sensor networks using a multivariate scheme," *Infocom 2006*, 2006.

[27] Diestel, R., *Graph Theory*, vol. 173 of *Graduate Texts in Mathematics*. Springer Verlag, third ed., 2005.

[28] Diffie, W. and Hellman, M., "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. IT-22, 1976.

[29] DiPietro, R., Mancini, L. V., and Mei, A., "Random key-assignment for secure wireless sensor networks," *Proceedings of the 1st ACM Workshop Security of Ad Hoc and Sensor Networks*, 2003.

[30] Douceur, J. R., "The sybil attack," *First International Workshop on Peer-to-Peer Systems (IPTPS '02)*, March 2002.

[31] Dousse, O., Mannersalo, P., and Thiran, P., "Latency of wireless sensor networks with uncoordinated power saving mechanisms," *The ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobHoc 2004)*, May 2004.

[32] Du, W., Deng, J., Han, Y. S., and Varshney, P. K., "A pairwise key predistribution scheme for wireless sensor networks," *10th ACM Conference on Computer and Communcations Security*, October 2003.

[33] Du, W., Deng, J., Han, Y. S., and Varshney, P. K., "A key management scheme for wireless sensor networks using deployment knowledge," *INFOCOM 2004*, March 2004.

[34] Eisenbud, D., *Commutative algebra with a view toward algebraic geometry*, vol. 150 of *Graduate Texts in Mathematics*. Springer-Verlag, 1995.

[35] Eschenauer, L. and Gligor, V., "A key-management scheme for distributed sensor networks," *ACM Conference on Computer and Communications Security*, 2002.

[36] Estrin, D., Govindan, R., and Heidemann, J., "Next century challenges: Scalable coordination in sensor networks," *ACM/IEEE International Conference on Mobile Computing and Networks (MobiCom '99)*, August 1999.

[37] Fekri, F., *Finite-field wavelet transforms and their application to error-control coding*. PhD thesis, Georgia Institute of Technology, July 2000.

[38] Fekri, F., Mersereau, R. M., and Schafer, R. W., "Theory of wavelet transforms over finite fields," in *Proc. Int. Conf. Acoust. Speech, and Signal proc.*, pp. 605–608, March 1999.

[39] Fekri, F., Mersereau, R. M., and Schafer, R. W., "Realization of paraunitary filter banks over fields of characteristic two," in *Proc. Int. Conf. Acoust. Speech, and Signal proc.*, June 2000.

[40] Fekri, F., Mersereau, R. M., and Schafer, R. W., "Theory of paraunitary filter banks over fields of characteristic two," *IEEE Trans. on Information Theory*, vol. 48, pp. 2964–2979, Nov. 2002.

[41] Fekri, F., Mersereau, R. M., and Schafer, R. W., "Two-band wavelets and filter banks over finite fields with connections to error control coding," *IEEE Transactions on Signal Processing*, vol. 51, pp. 3143–3151, December 2003.

[42] Feldman, F., *Fast Spectral Tests for Measuring Nonrandomness and the DES*. Springer-Verlag, 1994.

[43] Gupta, P. and Kumar, P., "The capacity of wireless networks," *IEEE Transactoins on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000.

[44] Gupta, P. and Kumar, P., "Critical power for asymptotic connectivity in wireless networks," *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W.M. McEneaney, G. Yin and Q. Zhang (Eds.)*, 1998.

[45] Heinzelman, W., Chandrakasan, A., and Balakrishnan, H., "Energy-efficient communication protocol for wireless sensor networks," *Proceedings of the Hawaii International Conference System Sciences*, January 2000.

[46] Heinzelman, W. B., Chandrakasan, A. P., and Balakrishnan, H., "An application specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, pp. 660–670, October 2002.

[47] Intanagonwiwat, C., Govindan, R., and Estrin, D., "Directed diffusion: A scalable and robust communication paradigm for sensor networks," *Mobile Computing and Networking*, pp. 56–67, 2000.

[48] Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., and Silva, F., "Directed diffusion for wireless sensor networking," *IEEE/ACM Transactions on networking (TON)*, pp. 2–16, 2003.

[49] Jakobsen, T. and Knudsen, L. R., "The interpolation attack on block ciphers," *Fast Software Encryption*, vol. LNCS 1267, pp. 28–40, 1997.

[50] Junior, W. R. P., de Paula Figueiredo, T. H., Wong, H. C., and Loureiro, A. A., "Malicious node detection in wireless sensor networks," *Proceedings fo the 18th International Parallel and Distributed Processing Symposium (IPDPS'04)*, 2004.

[51] Kahn, J., Katz, R., and Pister, K., "Next century challenges: Mobile networking for 'Smart Dust'," *International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 271–278, 1999.

[52] Karlof, C. and Wagner, D., "Secure routing in wireless sensor networks: Attacks and countermeasures," *IEEE International Workshop on Sensor Network Procotols and Applications*, 2003.

[53] Karp, B. and Hung, H. T., "GPSR: Greedy perimeter stateless routing for wireless networks," *Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 2000.

[54] Kerckhoffs, A., "La cryptographie militaire," *Journal des sciences militaires*, vol. IX, pp. 5–83, Jan. 1883.

[55] Kipnis, A. and Shamir, A., "Cryptanalysis of the HFE public key cryptosystem by relinearization," *CRYPTO99*, pp. 19–30, 1999.

[56] Koc, C. K. and Acar, T., "Montgomery multiplication in $\mathbb{GF}(2^k)$," *Designs, Codes and Cryptography*, vol. 14, pp. 57–69, April 1998.

[57] Lee, S., Bhattacharjee, B., and Banerjee, S., "Efficient geographic routing in multihop wireless networks," *The ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc 2005)*, May 2005.

[58] Lidl, R. and Niederreiter, H., *Finite Fields*. Addison-Wesley Publishing Company, 1983.

[59] Lindsey, S. and Raghavendra, C. S., "PEGASIS: Power efficient gathering in sensor information systems," *IEEE Aerospace Conference*, March 2002.

[60] Liu, D. and Ning, P., "Establishing pairwise keys in distributed sensor networks," *Proceedings for the 10th ACM Conference on Computer and Communication Security*, pp. 52–61, 2003.

[61] Liu, D. and Ning, P., "Location based pairwise key establishment for static sensor networks," *2003 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03)*, October 2003.

[62] Manjeshwar, A. and Agrawal, D. P., "TEEN: A routing protocol for enhanced efficiency in wireless sensor networks," *IPDPS 2001*, 2001.

[63] Manjeshwar, A. and Agrawal, D. P., "APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks," *Proceedings of the International Parallel and Distributed Processing Symposium*, 2002.

[64] Marti, S., Giuli, T. J., Lai, K., and Baker, M., "Mitigating routing misbehavior in mobile ad hoc networks," *Sixth annual ACM/IEEE Internation Conference on Mobile Computing and Networking*, pp. 225–265, 2000.

[65] Massey, J. L., "The discrete-fourier transform in coding and cryptography," in *Proc. ITW*, San Diego, CA, Feb. 1998.

[66] Massey, J. L. and Serconek, S., *A Fourier transform approach to the linear complexity of nonlinearly filtered sequences*. LNCS 839, 1994.

[67] Matsui, M., *Linear Cryptanalysis Method for DES Cipher*. LNCS 765, 1994.

[68] Miyaguchi, S., "The feal-8 cryptosystem and a call for attack," *CRYPTO*, pp. 624–627, 1989.

[69] Moscibroda, T., von Rickenbach, P., and Wattenhofer, R., "Analyzing the energy-latency trade-off during the deployment of sensor networks," *The 25th Conference on Computer Communications (INFOCOM 2006)*, April 2006.

[70] National Bureau of Standards, *Federal Information Processing Standards Publication 46-2*, January 1988.

[71] Newsome, J., Shi, E., Song, D., and Perrig, A., "The sybil attack in sensor networks: Analysis and defenses," *Third International Symposium on Information Processing in Sensor Networks (IPSN 2004)*, 2004.

[72] Niculescu, D. and Nath, B., "Trajectory based forwarding and its applications," *The Ninth Annual International Conference on Mobile Computing and Networking (Mobicom 2003)*, September 2003.

[73] NIST, "Recommendation for the triple data encryption algorithm (tdea) block cipher," *Special Publication 800-67*, May 2004.

[74] Noorkami, M. and Fekri, F., "An efficient convolution to enhance the performance of the wavelet cryptosystem on handheld devices," *International Conference on Information Technology: Coding and Computing*, pp. 495–500.

[75] Parno, B., Perrig, A., and Gligor, V., "Distributed detection of node replication attacks in sensor networks," *IEEE Symposium on Security and Privacy*, 2005.

[76] Perrig, A., Szewczyk, R., Wen, V., Culler, D. E., and Tygar, J. D., "SPINS: security protocols for sensor networks," *Mobile Computing and Networking*, pp. 189–199, 2001.

[77] Phoong, S.-M., Kim, C., Vaidyanathan, P., and Ansari, R., "A new class of two-channel biorthogonal filter banks and waveletbases," *IEEE Transactions on Signal Processing*, vol. 43, pp. 649–665, March 1995.

[78] Pirzada, A. A. and McDonald, C., "Circumventing sinkholes and wormholes in. wireless sensor networks," *2005 International Workshop on Wireless Ad-hoc Networks*, May 2005.

[79] Pishro-Nik, H., Chan, K. S., and Fekri, F., "On connectivity properties of large-scale sensor networks," *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004*, pp. 498–507, 2004.

[80] Pishro-Nik, H., Chan, K. S., and Fekri, F., "On connectivity properties of large-scale sensor networks," *accepted to Wireless Networks, Kluwer Academic Press*, 2005.

[81] Preneel, B., Rompay, B. V., Ors, S. B., Biryukov, A., Granboulan, L., Dottax, E., Dichtl, M., Schafheutle, M., Serf, P., Pyka, S., Biham, E., Barkan, E., Dunkelman, O., Stolin, J., Ciet, M., Quisquater, J.-J., Sica, F., Raddum, H., and Parker, M., "NESSIE D21 - performance of optimized implementations of the NESSIE primitives," 2003.

[82] Qayyum, A., Viennot, L., and Laouiti, A., "Multipoint relaying for flooding broadcast messages in mobile wireless networks," *Proceedings of the 35th Hawaii International Conference on System Sciences*, 2002.

[83] Qian, L., Song, N., and Li, X., "Detecting and locating wormhole attacks in wireless ad hoc networks through statistical analysis of multi-path," *Wireless Communications and Networking Conference (WCNC 2005)*, March 2005.

[84] Rao, A., Ratnasamy, S., Papadimitriou, C., Shenker, S., and Stoica, I., "Geographic routing without location information," *The Ninth Annual International Conference on Mobile Computing and Networking (MobiCom'03)*, September 2003.

[85] Rivest, R., Shamir, A., and Adleman, L., "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[86] Rivest, R., "The rc5 encryption algorithm," *In the Proceedings of the Second International Workshop on Fast Software Encryption (FSE)*, pp. 86–96, 1994.

[87] Rueppel, R., *Analysis and design of stream ciphers*. Springer-Verlag, N.Y., 1986.

[88] Santos, R., Edwards, A., Alvarez, O., Gonzalez, A., and Verduzco, A., "A geographic routing algorithm for wireless sensor networks," *Elecronics, Robotics and Automotive Mechanics Conference*, vol. 1, pp. 64–69, September 2006.

[89] Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., and Ferguson, N., *The Twofish Encryption Algorithm*. Wiley, 1998.

[90] Seada, K., Zuniga, M., Helmy, A., and Krishnamachari, B., "Energy-efficient forwarding strategies for geographic routing in lossy wireless sensor networks," *SenSys '04*, November 2004.

[91] Shannon, C. E., "Communication theory of secrecy systems," *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[92] Sohrabi, K., Gao, J., Ailawadhi, V., and Pottie, G., "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, October 2000.

[93] Staddon, J., Balfanz, D., and Durfee, G., "Efficient tracing of failed nodes in sensor networks," *Proceedings of the first ACM international workshop on Wireless sensor networks and applications*, pp. 122–130, September 2002.

[94] Stallings, W., *Network Security Essentials: Applications and Standards*. Prentice-Hall, 199.

[95] Stanley, R. P., *Enumerative Combinatorics*, vol. 1. Cambridge University Press, 1997.

[96] Subramamian, R., Chan, K., and Fekri, F., "Analysis of latency in secure wireless sensor networks with key predistribution," *to appear in the Asilomar Conference on Signals, Systems, and Computers (invited paper)*, October 2008.

[97] W.Diffie and M.E.Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 6, pp. pp.644–654, 1976.

[98] Webster, A. F. and Tavares, S. E., "On the design of s-boxes," *CRYPTO*, pp. 523–534, 1985.

[99] Xue, F. and Kumar, P. R., "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, no. 2, pp. 169–181, 2004.

[100] Ye, W., Heidemann, J., and Estrin, D., "An energy-efficient mac protocol for wireless sensor networks," *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications*, June 2002.

[101] Yu, Y., Krishnamachari, B., and Prasanna, V., "Energy-latency tradeoffs for data gathering in wireless sensor networks," *The 23rd Conference on Computer Communications (INFOCOM 2004)*, March 2004.

[102] Zhou, L. and Haas, Z. J., "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, November/December 1999.

[103] Zorzi, M. and Rao, R., "Energy and latency performance of geographic random forwarding for ad hoc and sensor network," *Wireless Communications and Networking (WCNC'03)*, March 2003.

VITA

Kevin S. Chan was born in Royal Oak, Michigan in 1979. He attended Hill Elementary School, Larson Middle School, and Troy Athens High School (Troy, MI) and graduated in June 1997. He attended college at Carnegie Mellon University (Pittsburgh, PA), where he received a Bachelor of Science degree in Electrical and Computer Engineering and Engineering and Public Policy in May 2001. Then, he received his Masters of Science degree in Electrical and Computer Engineering from the Georgia Institute of Technology (Atlanta, GA) in May 2003. In 2005, he worked as a student intern in the Special Projects group in Tactical Electronic Warfare Division at the United States Naval Research Laboratory (Washington, DC). In his graduate studies at the Georgia Institute of Technology, his advisor was Dr. Faramarz Fekri. His research interests include cryptography and security services for wireless sensor networks.